Foundations of Probability in R

Samuel Ab. Gebremariam

2024-10-06

Calculating probabilities

You're in charge of the sales team, and it's time for performance reviews, starting with Amir. As part of the review, you want to randomly select a few of the deals that he's worked on over the past year so that you can look at them more deeply. Before you start selecting deals, you'll first figure out what the chances are of selecting certain deals.

Instructions

- Count the number of deals Amir worked on for each product type.
- Create a new column called prob by dividing n by the total number of deals Amir worked on.
- QUESTION 1: If you randomly select one of Amir's deals, what's the probability that the deal will involve Product C?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# amir_deals <- read.csv(file.choose(), header = T)
amir_deals <- read.csv("amir_deals.csv")
head(amir_deals, 10)
```

```
##     X   product  client status  amount num_users
## 1   1 Product F Current    Won 7389.52        19
## 2   2 Product C     New    Won 4493.01        43
## 3   3 Product B     New    Won 5738.09        87
## 4   4 Product I Current    Won 2591.24        83
## 5   5 Product E Current    Won 6622.97        17
## 6   6 Product B     New    Won 5496.27         2
## 7   7 Product C Current    Won 3043.13        29
## 8   8 Product N Current    Won 7340.64        13
## 9   9 Product F Current    Won 6780.85        80
## 10 10 Product B Current    Won 5237.24        23
```

```
# Count the deals for each product
amir_deals %>%
  count(product)
```

```
##      product  n
## 1  Product A 23
## 2  Product B 62
## 3  Product C 15
## 4  Product D 40
## 5  Product E  5
## 6  Product F 11
## 7  Product G  2
## 8  Product H  8
## 9  Product I  7
## 10 Product J  2
## 11 Product N  3
```

```
# Calculate probability of picking a deal with each product
amir_deals %>%
  count(product) %>%
  mutate(prob = n/sum(n))
```

```
##      product  n       prob
## 1  Product A 23 0.12921348
## 2  Product B 62 0.34831461
## 3  Product C 15 0.08426966
## 4  Product D 40 0.22471910
## 5  Product E  5 0.02808989
## 6  Product F 11 0.06179775
## 7  Product G  2 0.01123596
## 8  Product H  8 0.04494382
## 9  Product I  7 0.03932584
## 10 Product J  2 0.01123596
## 11 Product N  3 0.01685393
```

Sampling deals

In the previous exercise, you counted the deals Amir worked on. Now it's time to randomly pick five deals so that you can reach out to each customer and ask if they were satisfied with the service they received. You'll try doing this both with and without replacement.

Additionally, you want to make sure this is done randomly and that it can be reproduced in case you get asked how you chose the deals, so you'll need to set the random seed before sampling from the deals.

dplyr is loaded and amir_deals is available.

Instructions

- Set the random seed to 31
- Take a sample of 5 deals without replacement
- Take a sample of 5 deals with replacement.
- QUESTION 2: What type of sampling is better to use for this situation?

```
# Set random seed to 31
set.seed(31)
```

```
# Sample 5 deals without replacement
amir_deals %>%
  sample_n(5)
```

```
##     X   product  client status  amount num_users
## 1 173 Product D Current   Lost 3086.88        55
## 2  49 Product C Current   Lost 3727.66        19
## 3  75 Product D Current   Lost 4274.80         9
## 4 168 Product B Current    Won 4965.08         9
## 5  43 Product A Current    Won 5827.35        50
```

```
# Set random seed to 31
set.seed(31)

# Sample 5 deals with replacement
amir_deals %>%
  sample_n(5, replace = TRUE)
```

```
##     X   product  client status  amount num_users
## 1 173 Product D Current   Lost 3086.88        55
## 2  49 Product C Current   Lost 3727.66        19
## 3  75 Product D Current   Lost 4274.80         9
## 4 168 Product B Current    Won 4965.08         9
## 5  43 Product A Current    Won 5827.35        50
```

```
  #sample_n(5)
```

Creating a probability distribution

A new restaurant opened a few months ago, and the restaurant's management wants to optimize its seating space based on the size of the groups that come most often. On one night, there are 10 groups of people waiting to be seated at the restaurant, but instead of being called in the order they arrived, they will be called randomly. In this exercise, you'll investigate the probability of groups of different sizes getting picked first. Data on each of the ten groups is contained in the restaurant_groups data frame.

Remember that expected value can be calculated by multiplying each possible outcome with its corresponding probability and taking the sum. The restaurant_groups data is available and dplyr and ggplot2 are loaded.

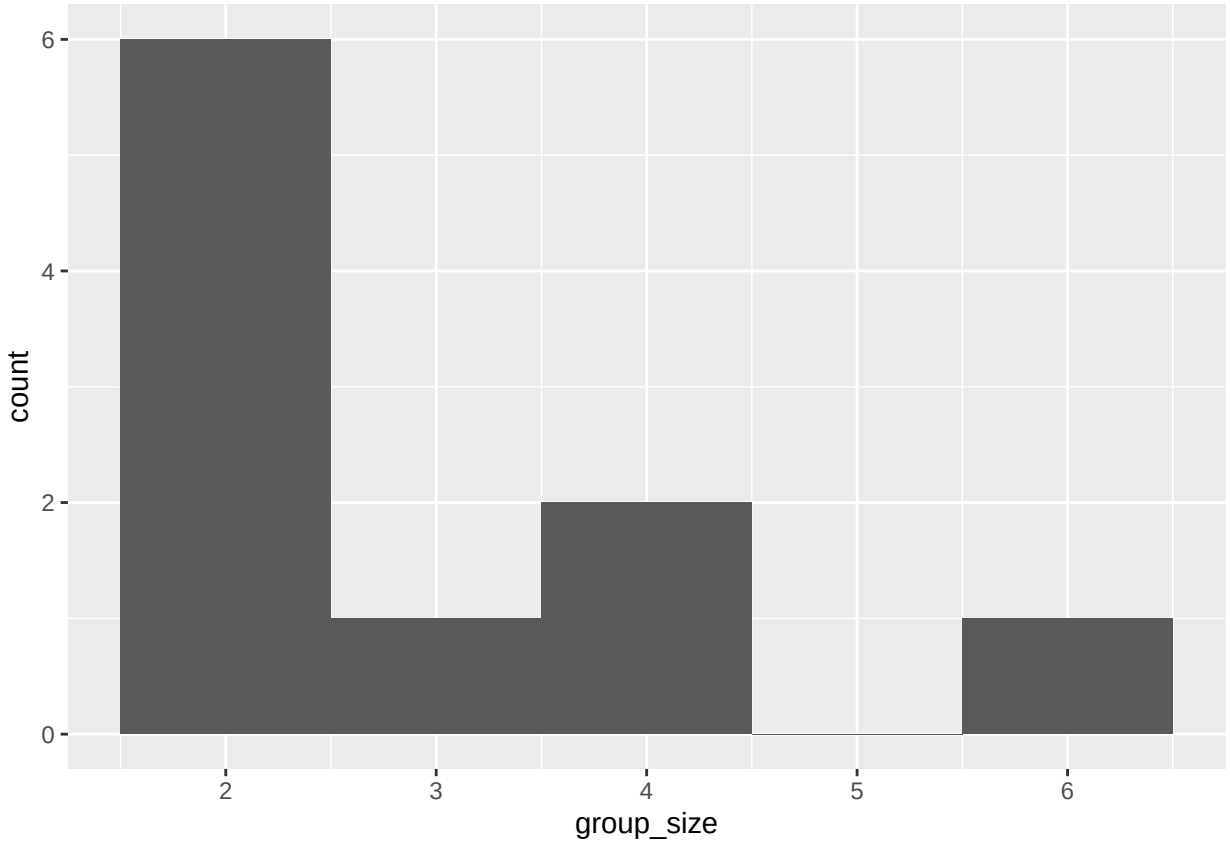
Instructions

- Create a histogram of the group_size column of restaurant_groups, setting the number of bins to 5.
- Count the number of each group_size in restaurant_groups, then add a column called probability that contains the probability of randomly selecting a group of each size. Store this in a new data frame called size_distribution.
- Calculate the expected value of the size_distribution, which represents the expected group size.
- Calculate the probability of randomly picking a group of 4 or more people by filtering and summarizing.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
restaurant_groups <- read.csv("restaurant_groups.csv")
# Create a histogram of restaurant_groups
ggplot(restaurant_groups, aes(group_size)) +
  geom_histogram(bins = 5)
```




```
# Create probability distribution
size_distribution <- restaurant_groups %>%
  # Count number of each group size
  count(group_size) %>%
  # Calculate probability
  mutate(probability = n / sum(n))

size_distribution
```

```
##   group_size n probability
## 1          2 6         0.6
## 2          3 1         0.1
## 3          4 2         0.2
## 4          6 1         0.1
```

```
# Create probability distribution
size_distribution <- restaurant_groups %>%
  count(group_size) %>%
  mutate(probability = n / sum(n))

# Calculate expected group size
expected_val <- sum(size_distribution$group_size *
                    size_distribution$probability)
expected_val
```

```
## [1] 2.9
```

```
# Create probability distribution
size_distribution <- restaurant_groups %>%
  count(group_size) %>%
  mutate(probability = n / sum(n))

# Calculate probability of picking group of 4 or more
size_distribution %>%
  # Filter for groups of 4 or larger
  filter(group_size >= 4) %>%
  # Calculate prob_4_or_more by taking sum of probabilities
  summarise(prob_4_or_above = sum(probability))
```

```
##   prob_4_or_above
## 1             0.3
```

```
restaurant_groups%>%
  count(group_size)%>%
  mutate(prob = n/sum(n)) %>%
  select(-n)
```

```
##   group_size prob
## 1          2  0.6
## 2          3  0.1
## 3          4  0.2
## 4          6  0.1
```

```
size_distribution <- restaurant_groups %>%
  count(group_size) %>%
  mutate(probability = n / sum(n)) %>%
  select(-n)
size_distribution
```

```
##   group_size probability
## 1          2         0.6
## 2          3         0.1
## 3          4         0.2
## 4          6         0.1
```

```
# Calculate probability of picking group of 4 or more
x=size_distribution %>%
  # Filter for groups of 4 or larger
  filter(group_size >= 4)
  # Calculate prob_4_or_more by taking sum of probabilities
  sum(x$probability)
```

```
## [1] 0.3
```

Data back-ups

The sales software used at your company is set to automatically back itself up, but no one knows exactly what time the back-ups happen. It is known, however, that back-ups happen exactly every 30 minutes. Amir comes back from sales meetings at random times to update the data on the client he just met with. He wants to know how long he'll have to wait for his newly-entered data to get backed up. Use your new knowledge of continuous uniform distributions to model this situation and answer Amir's questions.

Instructions

- To model how long Amir will wait for a back-up using a continuous uniform distribution, save his lowest possible wait time as min and his longest possible wait time as max. Remember that back-ups happen every 30 minutes.

```
# Min and max wait times for back-up that happens every 30 min
min <- 0
max <- 30
```

- Calculate the probability that Amir has to wait less than 5 minutes, and store in a variable called prob_less_than_5.

```
# Min and max wait times for back-up that happens every 30 min
min <- 0
max <- 30

# Calculate probability of waiting less than 5 mins
prob_less_than_5 <- punif(5, min, max)
prob_less_than_5
```

```
## [1] 0.1666667
```

```
?punif()
```

```
## starting httpd help server ... done
```

- Calculate the probability that Amir has to wait more than 5 minutes, and store in a variable called prob_greater_than_5.

```
# Min and max wait times for back-up that happens every 30 min
min <- 0
max <- 30

# Calculate probability of waiting more than 5 mins
prob_greater_than_5 <- punif(5, min, max, lower.tail = FALSE)
prob_greater_than_5
```

```
## [1] 0.8333333
```

- Calculate the probability that Amir has to wait between 10 and 20 minutes, and store in a variable called prob_between_10_and_20.

```
# Min and max wait times for back-up that happens every 30 min
min <- 0
max <- 30

# Calculate probability of waiting 10-20 mins
prob_between_10_and_20 <- punif(20, min, max) - punif(10, min, max)
prob_between_10_and_20
```

```
## [1] 0.3333333
```

Simulating wait times

To give Amir a better idea of how long he'll have to wait, you'll simulate Amir waiting 1000 times and create a histogram to show him what he should expect. Recall from the last exercise that his minimum wait time is 0 minutes and his maximum wait time is 30 minutes.

A data frame called wait_times is available and dplyr and ggplot2 are loaded

- Set the random seed to 334

```
# Set random seed to 334
set.seed(334)
```

- Generate 1000 wait times from the continuous uniform distribution that models Amir's wait time. Add this as a new column called time in the wait_times data frame.

```
# Set random seed to 334
set.seed(334)

# Generate 1000 wait times between 0 and 30 mins, save in time column
wait_times <- read.csv("wait_times.csv")
wait_times %>%
  mutate(time = runif(1000, min = 0, max = 30))
```

```
##      simulation_nb        time
## 1                1 29.45315459
## 2                2 16.29111785
## 3                3  0.36924658
## 4                4 24.18677361
## 5                5 23.42601313
## 6                6 15.92492170
## 7                7 17.54207142
## 8                8  2.81082236
## 9                9  1.22953709
## 10              10 18.81465174
## 11              11 15.84175237
## 12              12  9.41775125
## 13              13  8.83677828
## 14              14 23.96499244
## 15              15 13.07967382
## 16              16  4.01695817
## 17              17  7.99449634
## 18              18  4.58148719
## 19              19 14.70777321
## 20              20 25.78698850
## 21              21 29.99619469
## 22              22 18.16970525
## 23              23 14.09443719
## 24              24 12.61347899
## 25              25 12.29471693
## 26              26  1.71292025
## 27              27 16.78696537
## 28              28 17.82353542
## 29              29  5.68281747
## 30              30 22.49724553
## 31              31 11.25188137
## 32              32 19.08385771
## 33              33  0.05211500
## 34              34  0.29523587
## 35              35 24.78207979
## 36              36 24.27452489
## 37              37 12.58902050
## 38              38  7.31610345
## 39              39 21.98134186
```

```
## 40              40  1.44244587
## 41              41 16.48523526
## 42              42 18.47448285
## 43              43 28.68268560
## 44              44 16.51950021
## 45              45 26.10806020
## 46              46 25.44633436
## 47              47 27.31116007
## 48              48 16.78878267
## 49              49  4.59194899
## 50              50 25.07935484
## 51              51 19.51173451
## 52              52 23.75916684
## 53              53 22.00560742
## 54              54 23.85031574
## 55              55  5.17543838
## 56              56  5.22998160
## 57              57 15.10675023
## 58              58 19.09363108
## 59              59 16.93706591
## 60              60 22.65432542
## 61              61 23.89705532
## 62              62  8.72819969
## 63              63  8.44010386
## 64              64 28.33278771
## 65              65 21.83944697
## 66              66 10.93783317
## 67              67 27.76403802
## 68              68  2.56605274
## 69              69  1.22772286
## 70              70 26.77751457
## 71              71 16.04902673
## 72              72 11.37484696
## 73              73 17.39148053
## 74              74 21.88181105
## 75              75  2.09128249
## 76              76  2.31436525
## 77              77  8.61378860
## 78              78 18.21443749
## 79              79 14.50121775
## 80              80  7.34539799
## 81              81 25.34504854
## 82              82 22.62102517
## 83              83 17.53644251
## 84              84 21.02231852
## 85              85 26.09942971
## 86              86 13.48281613
## 87              87 21.73122370
## 88              88 24.38144228
## 89              89 29.51306323
## 90              90 26.85805804
## 91              91  7.09091901
## 92              92 13.55704284
## 93              93  4.64813914
```

```
## 94              94 11.72299788
## 95              95 29.16724531
## 96              96  2.56408873
## 97              97 19.96485271
## 98              98 26.84800347
## 99              99  1.98970109
## 100            100 27.80477462
## 101            101  4.68890270
## 102            102 29.75399736
## 103            103 28.86156492
## 104            104  1.62335933
## 105            105 26.58146717
## 106            106  6.95498544
## 107            107 24.37387391
## 108            108 20.63972661
## 109            109  3.31054327
## 110            110 15.24809941
## 111            111 16.66182927
## 112            112 21.08719027
## 113            113  2.13087585
## 114            114  5.50382916
## 115            115  2.18534266
## 116            116 16.72306163
## 117            117 15.27638286
## 118            118 21.93987240
## 119            119  8.14819436
## 120            120 13.22942961
## 121            121  4.81524780
## 122            122  4.63081524
## 123            123  3.90025639
## 124            124 12.27260683
## 125            125 13.50092378
## 126            126  8.49478288
## 127            127  1.31813536
## 128            128 18.35409017
## 129            129 19.33882381
## 130            130  3.67552059
## 131            131 10.96036061
## 132            132 17.36368593
## 133            133  9.11544772
## 134            134 20.69915994
## 135            135 27.80066973
## 136            136  9.96377338
## 137            137 24.10210552
## 138            138  8.26637429
## 139            139 22.07137189
## 140            140 15.21410348
## 141            141  7.70470032
## 142            142 24.57876420
## 143            143  5.11461786
## 144            144 20.73345027
## 145            145  8.30975461
## 146            146 16.99033177
## 147            147 26.57024723
```

```
## 148           148 18.91592598
## 149           149 20.11793115
## 150           150 25.66186169
## 151           151  3.30988693
## 152           152 12.44513940
## 153           153  3.90224391
## 154           154 13.82439813
## 155           155 14.13531585
## 156           156 13.10707968
## 157           157 21.15397943
## 158           158 10.74559600
## 159           159 22.55146616
## 160           160 21.56216869
## 161           161  1.36846875
## 162           162  4.44459092
## 163           163  0.47738479
## 164           164 15.44300649
## 165           165 13.61435918
## 166           166 14.59740393
## 167           167 18.13859917
## 168           168  4.90525077
## 169           169 24.78090435
## 170           170 18.75689236
## 171           171  4.70107029
## 172           172 25.96421457
## 173           173 21.05015198
## 174           174 20.60453370
## 175           175 28.19999088
## 176           176 27.77331149
## 177           177 22.88689508
## 178           178 23.77027513
## 179           179 19.82064425
## 180           180 13.67029054
## 181           181 19.86780181
## 182           182  5.09980166
## 183           183 20.09009174
## 184           184  9.19944863
## 185           185  3.47489387
## 186           186  7.87616526
## 187           187 15.94566443
## 188           188 10.17268332
## 189           189 17.05031343
## 190           190 27.05733098
## 191           191 27.02909490
## 192           192  2.83697048
## 193           193  7.85712880
## 194           194 18.33081177
## 195           195 10.23665127
## 196           196 26.60579928
## 197           197 23.83012579
## 198           198  9.70762531
## 199           199  3.94041099
## 200           200 22.12165447
## 201           201 10.68154077
```

```
## 202           202  0.76626179
## 203           203  7.09016587
## 204           204  4.05329900
## 205           205  0.83210992
## 206           206 18.85119532
## 207           207  4.35439821
## 208           208  0.64977399
## 209           209 25.84636275
## 210           210 15.84230244
## 211           211 25.86542755
## 212           212 26.07684359
## 213           213  6.18999966
## 214           214  8.43593902
## 215           215 24.18478795
## 216           216 10.62778520
## 217           217 17.19132731
## 218           218 11.31612540
## 219           219 27.80950904
## 220           220  7.08353750
## 221           221 22.54112792
## 222           222 22.89749492
## 223           223 14.78477028
## 224           224 14.04549201
## 225           225 14.93056498
## 226           226 15.61904760
## 227           227 27.90506173
## 228           228 29.03126386
## 229           229 17.95046012
## 230           230  9.50867706
## 231           231 28.96867423
## 232           232 17.23006020
## 233           233 16.48326953
## 234           234 29.62941040
## 235           235 29.53264363
## 236           236 13.06570023
## 237           237  5.51182835
## 238           238  8.97947283
## 239           239  6.95682119
## 240           240 13.94301978
## 241           241  5.88415202
## 242           242 29.99165510
## 243           243 27.53866999
## 244           244 17.52703829
## 245           245 28.10205803
## 246           246  7.26896494
## 247           247  5.85534172
## 248           248 19.06698948
## 249           249 25.44289668
## 250           250 13.32176088
## 251           251  6.08293406
## 252           252 10.95717730
## 253           253 15.72299918
## 254           254  6.05257920
## 255           255 10.13583530
```

```
## 256           256 16.32800652
## 257           257  0.50360160
## 258           258 24.69565599
## 259           259 19.79090953
## 260           260  6.41763006
## 261           261 21.39276288
## 262           262 15.25108414
## 263           263 16.11735773
## 264           264 24.61424236
## 265           265 24.08914387
## 266           266 14.65908080
## 267           267  1.53653435
## 268           268 21.94335671
## 269           269  7.29698789
## 270           270 25.79171011
## 271           271 13.81756713
## 272           272  1.52692696
## 273           273 28.68397699
## 274           274 24.44759800
## 275           275 18.47182235
## 276           276  9.82464750
## 277           277  7.37684553
## 278           278 27.34689845
## 279           279 11.81990997
## 280           280 18.56640825
## 281           281 28.41001900
## 282           282  3.69871438
## 283           283 21.62252435
## 284           284 29.01304065
## 285           285 21.17122008
## 286           286  9.23375448
## 287           287  9.28087743
## 288           288  9.11876739
## 289           289 14.28881688
## 290           290 27.64225040
## 291           291 29.12166760
## 292           292 14.81106031
## 293           293 19.26581445
## 294           294  2.18942036
## 295           295 13.09335056
## 296           296 19.36590429
## 297           297 20.83139085
## 298           298 14.47464931
## 299           299 22.82562831
## 300           300 27.00556817
## 301           301  6.12020946
## 302           302 26.50418573
## 303           303  2.33169044
## 304           304 18.21444613
## 305           305 17.73878159
## 306           306  0.88235375
## 307           307  3.94441641
## 308           308  7.16087096
## 309           309 19.83257173
```

```
## 310           310 21.29177334
## 311           311 14.23450629
## 312           312 15.29842467
## 313           313  2.49413843
## 314           314 19.13876231
## 315           315 23.24597251
## 316           316 17.85439791
## 317           317 16.83771461
## 318           318 10.20920923
## 319           319 10.30384726
## 320           320  2.80929160
## 321           321 26.55758000
## 322           322 29.83452890
## 323           323 10.94769653
## 324           324 13.64371708
## 325           325 27.14646269
## 326           326 11.73016779
## 327           327 23.42090358
## 328           328 18.68195778
## 329           329  6.77623506
## 330           330  4.32711476
## 331           331 26.58448611
## 332           332 20.53492705
## 333           333 21.78986641
## 334           334 19.78927551
## 335           335  9.16987331
## 336           336 25.49647434
## 337           337 12.95142716
## 338           338  9.21591724
## 339           339  3.71928454
## 340           340 10.18400854
## 341           341 13.81445620
## 342           342 18.75217533
## 343           343  3.04595439
## 344           344 10.79786948
## 345           345 29.31001355
## 346           346  1.21589832
## 347           347 16.71420172
## 348           348  7.01302179
## 349           349  2.50405112
## 350           350 14.31464099
## 351           351 23.06294401
## 352           352 14.79304570
## 353           353 17.88536563
## 354           354 10.09426212
## 355           355 28.05442481
## 356           356 21.08625339
## 357           357 17.73074598
## 358           358 22.54238734
## 359           359 18.68309828
## 360           360  9.12667959
## 361           361  6.78268570
## 362           362 12.87106037
## 363           363 13.27456597
```

```
## 364           364 11.96363262
## 365           365  7.82005662
## 366           366 14.76887760
## 367           367 25.22041113
## 368           368 28.75726938
## 369           369  4.62506805
## 370           370  8.21708702
## 371           371 14.27739504
## 372           372  5.19841835
## 373           373 29.14449930
## 374           374  7.17904563
## 375           375 21.33901682
## 376           376  9.15599429
## 377           377 29.91129147
## 378           378  7.24296600
## 379           379  0.56037860
## 380           380  2.84814615
## 381           381  1.77340397
## 382           382 21.86930791
## 383           383  2.43868958
## 384           384 13.71335877
## 385           385 22.65005260
## 386           386  2.35276919
## 387           387  6.99185851
## 388           388 16.07555050
## 389           389  1.66261677
## 390           390 18.19484275
## 391           391 19.43370478
## 392           392  8.45190485
## 393           393  5.64412612
## 394           394 27.98906704
## 395           395 22.63233759
## 396           396 25.22347289
## 397           397 13.63546245
## 398           398  7.64288333
## 399           399 15.19147471
## 400           400  2.91592826
## 401           401  7.02979135
## 402           402 17.54522154
## 403           403  9.56778450
## 404           404  2.31630396
## 405           405  9.02047624
## 406           406  9.19811845
## 407           407 29.74029722
## 408           408 27.20223204
## 409           409  5.07001682
## 410           410  6.42362910
## 411           411 10.36749214
## 412           412 17.06248441
## 413           413 14.09376311
## 414           414 20.99937669
## 415           415 19.15727428
## 416           416  6.19942550
## 417           417 18.63208443
```

```
## 418           418 11.16741173
## 419           419 12.83131198
## 420           420 14.85977447
## 421           421 13.66304567
## 422           422 19.83423738
## 423           423  4.01754856
## 424           424 27.47191604
## 425           425 23.92742818
## 426           426 18.87272294
## 427           427 26.66857070
## 428           428  0.15995986
## 429           429 16.22308373
## 430           430 13.42008427
## 431           431 16.17911689
## 432           432 10.01260702
## 433           433 28.56098204
## 434           434 25.75032657
## 435           435 21.39014875
## 436           436 17.99363716
## 437           437 19.80622008
## 438           438 28.30116014
## 439           439 21.51915263
## 440           440 18.16048590
## 441           441 20.55970985
## 442           442 29.42974746
## 443           443 21.53584968
## 444           444 14.64190344
## 445           445 17.29999723
## 446           446 27.82951849
## 447           447 18.69110554
## 448           448 21.12538267
## 449           449 15.30695465
## 450           450  3.55855007
## 451           451 25.98504815
## 452           452 18.26334755
## 453           453 22.07158551
## 454           454 29.04561422
## 455           455 11.31640266
## 456           456 21.28965942
## 457           457 22.34975370
## 458           458  0.52093525
## 459           459 20.34933524
## 460           460  1.42289677
## 461           461 27.59932949
## 462           462  2.89372426
## 463           463 18.81056723
## 464           464  6.40705688
## 465           465 21.20666586
## 466           466 15.50679774
## 467           467  6.70022594
## 468           468 27.89876780
## 469           469 19.11103416
## 470           470 14.01322254
## 471           471 16.22470100
```

```
## 472           472 27.87636896
## 473           473 24.34938096
## 474           474 25.39768851
## 475           475  4.45197798
## 476           476  1.53750490
## 477           477 22.63620186
## 478           478 14.25112401
## 479           479 24.80635514
## 480           480 15.37004618
## 481           481 21.88684908
## 482           482 10.51672163
## 483           483 24.50996733
## 484           484 11.72664362
## 485           485 18.45670694
## 486           486  4.31498132
## 487           487 12.07116565
## 488           488  5.01115491
## 489           489 22.04236976
## 490           490  4.63524594
## 491           491 22.42581264
## 492           492  2.26363320
## 493           493  6.90867981
## 494           494 26.44155943
## 495           495  0.55509018
## 496           496  0.14361411
## 497           497 15.38088849
## 498           498  8.41076879
## 499           499 28.90797283
## 500           500 23.83094005
## 501           501 18.23616864
## 502           502  2.37112971
## 503           503 27.59622060
## 504           504  2.43101197
## 505           505 20.11683017
## 506           506  1.15869150
## 507           507 21.94247100
## 508           508 21.83020435
## 509           509  9.59816972
## 510           510 18.86794155
## 511           511 26.48611553
## 512           512  2.29030104
## 513           513 15.46201328
## 514           514  7.95034215
## 515           515 17.05755436
## 516           516  6.58767968
## 517           517 10.07531458
## 518           518  1.78737030
## 519           519 14.13113301
## 520           520  5.33891652
## 521           521  7.99536390
## 522           522 17.49613540
## 523           523 29.61214882
## 524           524 22.00217134
## 525           525 10.62230572
```

```
## 526           526  5.39191182
## 527           527 28.28376311
## 528           528  4.09335926
## 529           529  3.32336883
## 530           530 13.09906285
## 531           531 12.29163133
## 532           532  1.02804122
## 533           533  3.63867698
## 534           534 28.89265722
## 535           535  1.05562723
## 536           536 27.72558298
## 537           537 24.31650326
## 538           538 15.21048615
## 539           539 23.74477396
## 540           540 10.95966802
## 541           541 14.82571477
## 542           542  9.69711231
## 543           543 13.49656616
## 544           544 29.05379775
## 545           545 26.99015575
## 546           546 11.70335405
## 547           547 24.72241084
## 548           548 20.80310749
## 549           549 21.29865146
## 550           550  4.27487598
## 551           551  0.18877497
## 552           552 15.52615040
## 553           553  8.03915869
## 554           554  7.31296097
## 555           555 15.91223379
## 556           556  1.51910824
## 557           557 23.18906535
## 558           558 15.88347173
## 559           559 16.51760637
## 560           560 23.22923348
## 561           561 15.25847444
## 562           562 25.09333366
## 563           563 13.93844577
## 564           564 15.25182184
## 565           565 18.29192387
## 566           566 16.18723450
## 567           567 15.78160366
## 568           568 18.97948554
## 569           569 27.56204371
## 570           570  6.90978096
## 571           571  2.81436247
## 572           572 22.30922569
## 573           573  4.22921974
## 574           574 25.49643823
## 575           575 25.86424883
## 576           576 15.46743162
## 577           577  9.50067313
## 578           578 28.23881332
## 579           579 17.41533428
```

```
## 580           580 26.07017294
## 581           581 11.27339201
## 582           582  0.44793117
## 583           583 17.17957113
## 584           584  0.57937812
## 585           585  4.70337945
## 586           586 15.36253812
## 587           587 24.09799757
## 588           588  9.98313663
## 589           589 24.29376919
## 590           590 20.03532447
## 591           591 16.86236385
## 592           592 28.49393641
## 593           593 24.80527276
## 594           594 10.29042019
## 595           595 17.28378473
## 596           596  4.29394705
## 597           597  4.13894035
## 598           598 27.10331543
## 599           599 19.01784361
## 600           600  4.13087844
## 601           601  7.61913192
## 602           602  0.22471917
## 603           603  7.58358095
## 604           604  0.70820356
## 605           605 26.15456688
## 606           606 25.00787396
## 607           607 12.13710360
## 608           608 29.51772553
## 609           609 23.29129434
## 610           610  0.79500558
## 611           611  8.64801754
## 612           612  8.92626902
## 613           613  3.55413855
## 614           614 20.95414013
## 615           615 13.97414454
## 616           616 12.05990767
## 617           617 25.30801366
## 618           618 21.94305107
## 619           619 27.24628801
## 620           620 26.99409339
## 621           621 21.48838394
## 622           622 25.64111639
## 623           623 16.02956112
## 624           624 24.97976664
## 625           625 19.43177693
## 626           626  4.07977849
## 627           627 22.07207988
## 628           628 18.20536372
## 629           629  8.25589764
## 630           630 25.94543448
## 631           631 26.68614750
## 632           632 13.17340614
## 633           633  4.13971917
```

```
## 634           634 18.69442663
## 635           635 21.15617341
## 636           636 26.62828154
## 637           637  1.58231570
## 638           638  6.39004376
## 639           639  1.36808490
## 640           640  7.59057867
## 641           641 12.24693146
## 642           642  5.26981595
## 643           643  5.19349064
## 644           644 16.23022058
## 645           645 19.43334169
## 646           646  8.32330740
## 647           647 21.81385769
## 648           648 26.36377704
## 649           649 16.52527308
## 650           650 21.36186962
## 651           651 21.76063207
## 652           652  9.43167117
## 653           653 28.81018798
## 654           654  1.97403420
## 655           655 20.11645230
## 656           656 20.45099184
## 657           657  8.20116245
## 658           658  7.61681224
## 659           659 22.39749865
## 660           660 20.45608649
## 661           661 23.24401018
## 662           662 16.03424410
## 663           663 24.69163685
## 664           664  7.58833783
## 665           665 19.79045543
## 666           666  4.83634152
## 667           667 22.71104205
## 668           668 22.46168953
## 669           669 21.28329621
## 670           670 14.84826955
## 671           671 17.71113721
## 672           672 24.99863414
## 673           673  1.11363730
## 674           674  5.18335772
## 675           675 17.93925157
## 676           676 15.98093777
## 677           677  9.71065524
## 678           678 28.08909436
## 679           679 12.12228267
## 680           680 21.41512543
## 681           681 27.59098297
## 682           682 10.43861453
## 683           683 17.17619657
## 684           684 15.64023849
## 685           685  7.81212623
## 686           686  7.94228212
## 687           687 27.67277035
```

```
## 688           688 11.51521985
## 689           689  7.23150658
## 690           690  3.21479284
## 691           691 19.54473338
## 692           692 15.11418222
## 693           693 29.11112360
## 694           694 18.85301092
## 695           695 24.07615190
## 696           696  4.43075927
## 697           697  8.07530223
## 698           698  7.10870020
## 699           699 16.42089552
## 700           700  5.14947769
## 701           701 26.45798509
## 702           702 15.92709096
## 703           703 22.54949688
## 704           704 23.11841410
## 705           705 23.20727685
## 706           706 24.02396576
## 707           707 28.11650863
## 708           708 23.53483649
## 709           709 10.83651507
## 710           710 14.52483727
## 711           711 19.27092228
## 712           712  4.96947936
## 713           713  6.37777295
## 714           714 19.66820788
## 715           715 10.33001991
## 716           716  2.76706099
## 717           717  2.65794648
## 718           718 13.18265599
## 719           719 15.87154296
## 720           720 26.15205297
## 721           721  3.47143965
## 722           722  5.68069562
## 723           723 21.81475559
## 724           724 18.99552750
## 725           725  4.07545613
## 726           726 22.48334356
## 727           727 25.15314901
## 728           728  4.14790113
## 729           729 26.80710363
## 730           730 12.71870700
## 731           731  0.23671982
## 732           732 18.27752029
## 733           733 20.38129024
## 734           734 25.49995270
## 735           735 19.34074980
## 736           736 27.88514764
## 737           737 23.62582308
## 738           738  4.57260252
## 739           739  4.84854108
## 740           740 13.21212999
## 741           741 27.19329578
```

```
## 742           742  1.45556869
## 743           743 24.08456709
## 744           744 16.55591665
## 745           745 10.49539158
## 746           746  2.12426508
## 747           747  3.30015942
## 748           748 27.21315860
## 749           749 13.60083407
## 750           750  6.42074840
## 751           751 15.80302568
## 752           752  7.64344939
## 753           753  3.61414381
## 754           754  3.44661766
## 755           755  4.44318358
## 756           756  2.85897752
## 757           757 18.31471199
## 758           758  9.24812254
## 759           759 15.63217551
## 760           760  6.64337234
## 761           761 25.32337808
## 762           762 10.41814572
## 763           763 19.13094874
## 764           764 22.86960758
## 765           765 22.85894301
## 766           766  5.86332334
## 767           767  0.05504268
## 768           768 27.66123282
## 769           769 25.70949992
## 770           770 29.95611676
## 771           771 15.84321597
## 772           772 10.35282277
## 773           773 12.69412339
## 774           774  3.80673125
## 775           775  6.16406827
## 776           776  2.99790168
## 777           777 24.26350489
## 778           778 16.80755850
## 779           779 13.83899124
## 780           780 10.87304362
## 781           781 10.24308386
## 782           782 10.72382789
## 783           783  9.30063335
## 784           784 10.87484150
## 785           785 10.25127639
## 786           786 19.50456381
## 787           787  1.03514625
## 788           788 24.25549979
## 789           789 13.27121684
## 790           790 25.87039498
## 791           791 11.24287691
## 792           792 15.81979402
## 793           793  3.38889360
## 794           794  7.21797730
## 795           795 21.59932912
```

```
## 796           796  1.33471450
## 797           797 17.68906805
## 798           798  0.33489309
## 799           799 13.29441077
## 800           800 25.58799972
## 801           801  8.34362524
## 802           802  5.24244705
## 803           803 29.86772894
## 804           804 11.27868217
## 805           805  1.77041067
## 806           806 12.26556163
## 807           807  4.08045416
## 808           808  9.23194574
## 809           809 12.71412474
## 810           810 21.38715362
## 811           811 12.41157442
## 812           812  3.27262945
## 813           813 17.68681702
## 814           814 21.29433260
## 815           815 12.17109138
## 816           816 10.58917779
## 817           817 16.98192174
## 818           818  2.54616702
## 819           819 25.22683462
## 820           820 16.03544157
## 821           821 15.04064024
## 822           822  3.77273770
## 823           823 18.32934281
## 824           824  5.93192280
## 825           825 10.68586400
## 826           826  2.71822324
## 827           827 14.42678049
## 828           828 11.41946887
## 829           829  3.00407069
## 830           830 10.19740361
## 831           831 21.18003317
## 832           832 26.14440167
## 833           833 17.06804041
## 834           834 23.59274075
## 835           835 11.64061781
## 836           836  2.32737881
## 837           837 20.44001259
## 838           838  4.76478560
## 839           839 24.71379337
## 840           840 24.77519453
## 841           841 17.99014271
## 842           842  5.75588196
## 843           843  2.16911993
## 844           844  1.22090104
## 845           845  3.82330611
## 846           846 20.15392916
## 847           847  9.60326056
## 848           848  5.16540379
## 849           849 12.45379922
```

```
## 850           850 13.28796059
## 851           851 11.86287006
## 852           852 20.29859643
## 853           853 13.84376860
## 854           854  4.95336537
## 855           855  9.14834554
## 856           856  6.23437267
## 857           857 12.74852165
## 858           858 18.42469239
## 859           859  2.50559419
## 860           860  4.44630169
## 861           861 27.09661371
## 862           862 21.36234795
## 863           863  3.00574969
## 864           864  0.83713321
## 865           865 25.36174874
## 866           866 28.87985375
## 867           867 13.74831506
## 868           868 25.35327182
## 869           869 12.70325953
## 870           870  5.17307359
## 871           871 13.63094065
## 872           872 14.55916693
## 873           873  6.27789009
## 874           874  2.80767876
## 875           875 22.85234640
## 876           876 19.30655676
## 877           877  0.27821728
## 878           878 16.68406505
## 879           879  6.40513663
## 880           880  7.72135927
## 881           881  0.79103180
## 882           882 27.04417687
## 883           883  1.79142114
## 884           884  5.56745985
## 885           885  2.18831601
## 886           886  2.21925403
## 887           887 17.70047102
## 888           888  6.21666149
## 889           889  1.36030485
## 890           890 21.21758044
## 891           891 23.19494686
## 892           892 14.47518775
## 893           893  8.84783751
## 894           894  5.32492983
## 895           895 22.48245830
## 896           896 19.28372380
## 897           897 27.48034459
## 898           898 24.70487057
## 899           899  8.00545029
## 900           900  8.45565304
## 901           901  7.80314541
## 902           902 25.80687525
## 903           903 28.20051021
```

```
## 904           904 13.05773278
## 905           905 19.17772947
## 906           906 29.72308018
## 907           907 20.27046272
## 908           908 17.97622324
## 909           909 21.67749231
## 910           910  1.12684628
## 911           911 22.94989524
## 912           912 24.13175539
## 913           913 28.41362476
## 914           914 25.11089469
## 915           915  0.67646472
## 916           916 16.55111630
## 917           917 15.80507174
## 918           918 21.53329839
## 919           919 27.35162175
## 920           920  4.24038327
## 921           921 15.61486362
## 922           922 11.89807023
## 923           923 21.03012824
## 924           924  8.61335917
## 925           925  2.95336662
## 926           926 22.10096980
## 927           927 21.42984963
## 928           928 26.57457779
## 929           929  0.32530567
## 930           930 11.67946577
## 931           931 24.84896766
## 932           932 28.30590534
## 933           933 19.27874641
## 934           934 22.59183099
## 935           935  5.54906694
## 936           936 15.92738783
## 937           937 23.82532625
## 938           938 27.78281898
## 939           939 11.12884755
## 940           940 17.37371000
## 941           941  2.08964465
## 942           942 14.29634497
## 943           943 17.93836294
## 944           944  0.02836663
## 945           945 23.52984892
## 946           946 12.08360466
## 947           947  4.83382977
## 948           948 22.58295289
## 949           949 24.41841301
## 950           950  4.35457614
## 951           951 18.55395347
## 952           952  9.81627590
## 953           953 12.85618000
## 954           954  3.75257013
## 955           955 14.58650635
## 956           956 28.88595521
## 957           957 12.81251761
```

```
## 958            958 17.77156874
## 959            959  2.36495243
## 960            960 10.50472292
## 961            961 10.88613254
## 962            962 18.03848800
## 963            963  6.85033504
## 964            964 29.35935741
## 965            965 14.77169513
## 966            966 15.44966381
## 967            967 11.82772621
## 968            968 28.18201558
## 969            969 28.28515080
## 970            970 25.84118795
## 971            971  2.32863119
## 972            972  6.74184497
## 973            973 15.22274302
## 974            974 13.61504895
## 975            975 26.69289316
## 976            976  2.79583029
## 977            977 20.07001006
## 978            978 28.03155079
## 979            979  5.06785383
## 980            980  2.62971973
## 981            981 28.59565169
## 982            982 11.50206714
## 983            983 11.71660302
## 984            984 15.06892989
## 985            985 15.64412930
## 986            986 23.11437869
## 987            987 12.91659115
## 988            988 21.31045686
## 989            989  8.83304000
## 990            990 28.03850106
## 991            991 24.60741165
## 992            992  1.44645144
## 993            993 26.57894109
## 994            994 29.12689109
## 995            995 23.14159813
## 996            996 24.09883305
## 997            997 26.35455905
## 998            998  9.82991366
## 999            999 20.18595314
## 1000          1000 22.39149915
```

```
head(wait_times, 10)
```

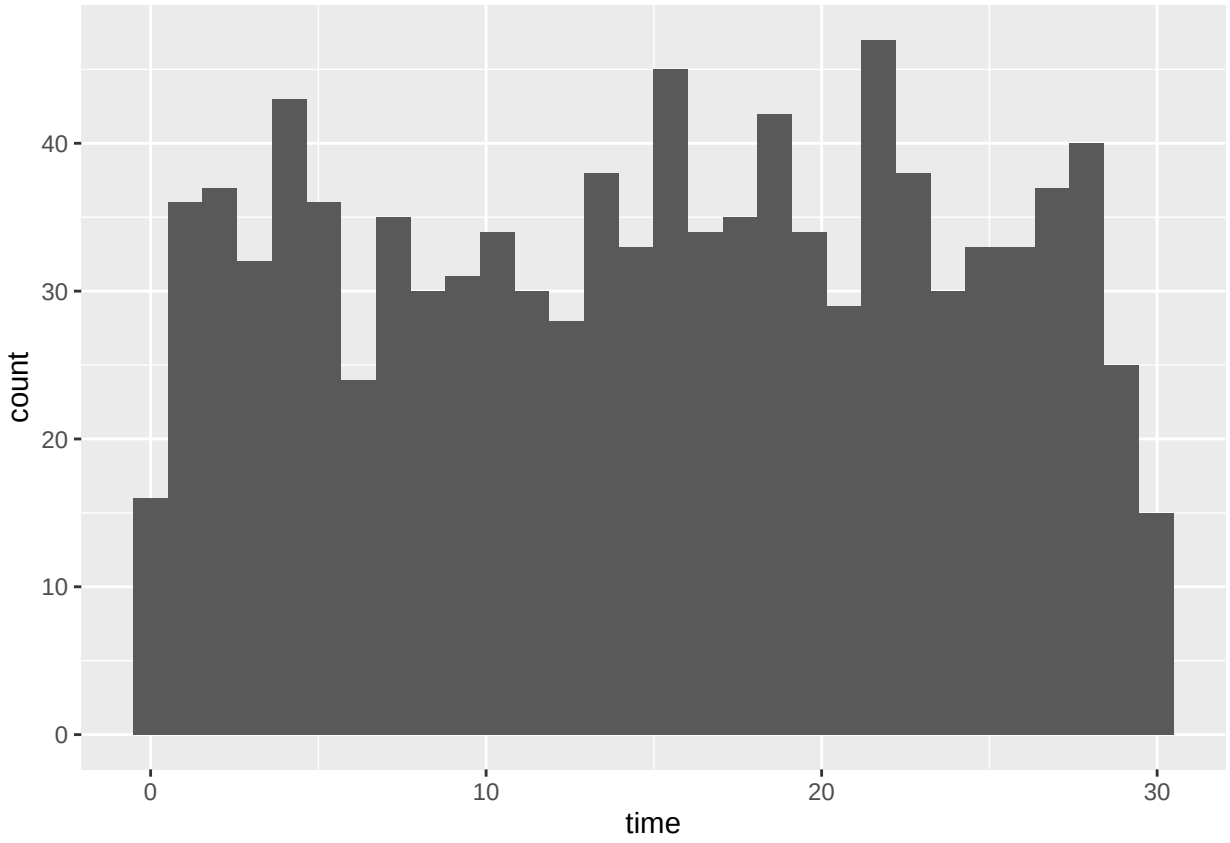
```
##    simulation_nb
## 1              1
## 2              2
## 3              3
## 4              4
## 5              5
## 6              6
## 7              7
## 8              8
```

```
## 9              9
## 10            10
```

- Create a histogram of the simulated wait times with 30 bins.

```
# Set random seed to 334
set.seed(334)

# Generate 1000 wait times between 0 and 30 mins, save in time column
wait_times %>%
  mutate(time = runif(1000, min = 0, max = 30)) %>%
  # Create a histogram of simulated times
  ggplot(aes(time)) +
  geom_histogram(bins = 30)
```



Simulating sales deals

Assume that Amir usually works on 3 deals per week, and overall, he wins 30% of deals he works on. Each deal has a binary outcome: it's either lost, or won, so you can model his sales deals with a binomial distribution. In this exercise, you'll help Amir simulate a year's worth of his deals so he can better understand his performance.

Instructions

- Set the random seed to 10 and simulate a single deal.

```
# Set random seed to 10
set.seed(10)
```

```r
# Simulate a single deal
rbinom(1, 1, 0.3)
```

```
## [1] 0
```

- Simulate a typical week of Amir's deals, or one week of 3 deals.

```r
# Set random seed to 10
set.seed(10)

# Simulate 1 week of 3 deals
rbinom(1, 3, 0.3)
```

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

- Simulate a year's worth of Amir's deals, or 52 weeks of 3 deals each, and store in deals. Calculate the mean number of deals he won per week.

```r
# Set random seed to 10
set.seed(10)

# Simulate 52 weeks of 3 deals
deals <- rbinom(52, 3, 0.3)

# Calculate mean deals won per week
mean(deals)
```

```
## [1] 0.8076923
```

```r
deals
```

```
##  [1] 1 0 1 1 0 0 0 0 1 1 1 1 0 1 1 1 0 0 1 2 2 1 1 1 1 1 2 0 1 1 1 0 0 2 1 1 2 2
## [39] 1 1 0 0 0 1 0 0 0 1 0 2 1 2
```

Calculating binomial probabilities

Just as in the last exercise, assume that Amir wins 30% of deals. He wants to get an idea of how likely he is to close a certain number of deals each week. In this exercise, you'll calculate what the chances are of him closing different numbers of deals using the binomial distribution.

Instructions

- What's the probability that Amir closes all 3 deals in a week?

```r
# Probability of closing 3 out of 3 deals
dbinom(3, 3, 0.3)
```

```
## [1] 0.027
```

- What's the probability that Amir closes 1 or fewer deals in a week?

```r
# Probability of closing <= 1 deal out of 3 deals
pbinom(1, 3, 0.3)
```

```
## [1] 0.784
```

- What's the probability that Amir closes more than 1 deal?

```r
# Probability of closing > 1 deal out of 3 deals
pbinom(1, 3, 0.3, lower.tail = FALSE)
```

```
## [1] 0.216
```

How many sales will be won?

Now Amir wants to know how many deals he can expect to close each week if his win rate changes. Luckily, you can use your binomial distribution knowledge to help him calculate the expected value in different situations. Recall from the video that the expected value of a binomial distribution can be calculated by $n \times p$

- Calculate the expected number of sales out of the 3 he works on that Amir will win each week if he maintains his 30% win rate.
- Calculate the expected number of sales out of the 3 he works on that he'll win if his win rate drops to 25%.
- Calculate the expected number of sales out of the 3 he works on that he'll win if his win rate rises to 35%.

```
# Expected number won with 30% win rate
won_30pct <- 3 * 0.3
won_30pct
```

```
## [1] 0.9
```

```
# Expected number won with 25% win rate
won_25pct <- 3 * 0.25
won_25pct
```

```
## [1] 0.75
```

```
# Expected number won with 35% win rate
won_35pct <- 3 * 0.35
won_35pct
```

```
## [1] 1.05
```