MA615 Assignment 1

Siwei Hu

September 15, 2018

Question:

A sloppy printer produces books with an average of 2 misprints per page. You want to know how many pages in a 50 page book will have more than k misprints. Make an n x k table that shows the probability that n or fewer pages in a 50 page book will have more than k misprints.

Idea:

The quesiton asked us about the probility that n or fewer pages in a 50 page book will have more than k misprints. So we first use poission distribution with λ = 2 to calculate the probability that each page have more than K misprints. The PDF of poission distribution is $P(x = k) = e^{-\lambda} * \frac{\lambda^k}{k!}$, so

$$P(x > k) = 1 - P(x \leq k) = 1 - \sum_{i=0}^{k} e^{-\lambda} * \frac{\lambda^k}{k!}$$

Then we got probabilities that k = 1:6 misprints each page and then we want to calculate probability that n or fewer page in a 50 page book will have more than k misprints. Since these pages are either bad pages or good pages, so we use binomial distribution to calculate these probabilities. The pdf of binomial is $P(x \leq n) = \binom{n}{k} * p^k * q^{n-k}$ We build a table as data frame and put all these probability into table

```
library(knitr)
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 3.4.4
```

```
# I set there are K misprint in each page of book
k <- c(1:6)

# use possion distribution with lamda = 2 to calculate the probability when misprint >= k in e
ach page

misprint.k <- ppois(k,2,lower.tail = FALSE)
misprint.k
```

```
## [1] 0.593994150 0.323323584 0.142876540 0.052653017 0.016563608 0.004533806
```

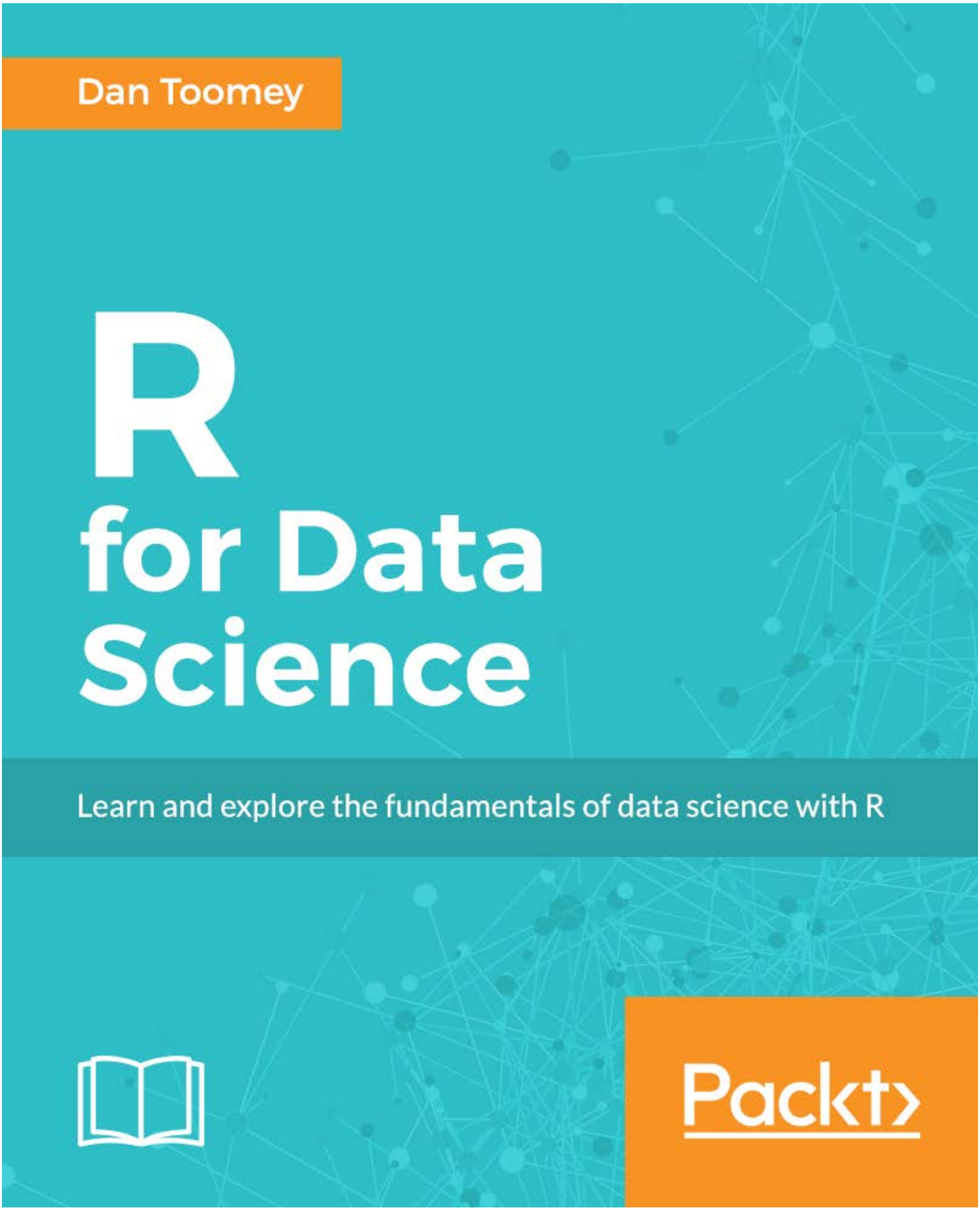
```
# build a table with size 50*6 and give these row and colume names to help understand
table <- as.data.frame(matrix(nrow= 50, ncol = 6,dimnames = list(c(1:50),c(1:6))))

#use a for loop and binomial distribution to calcaulate the probabilities that misprint >= k a
nd page <=n in each in 50 rows
```

```
  table[n,] <- pbinom(n,50, misprint.k,lower.tail = TRUE)
}
#Use Kable to set the whole table like adjusting probilities digits and showing row name or no
t.
table.1 = kable(x = table, digits = 10,row.names = TRUE,"html")
kable_styling(kable_input = table.1,bootstrap_options = "striped",font_size = 14)
```

	1	2	3	4	5	6
1	0.0000000000	0.0000000822	0.004190514	0.2528294	0.7991602	0.9781965
2	0.0000000000	0.0000010063	0.019471110	0.5060029	0.9499136	0.9984423
3	0.0000000000	0.0000080707	0.060225821	0.7311428	0.9905388	0.9999176
4	0.0000000000	0.0000477324	0.140049772	0.8781723	0.9985786	0.9999966
5	0.0000000000	0.0002220798	0.262465886	0.9533530	0.9998243	0.9999999
6	0.0000000000	0.0008468685	0.415510269	0.9846918	0.9999817	1.0000000
7	0.0000000000	0.0027233491	0.575867877	0.9956402	0.9999984	1.0000000
8	0.0000000003	0.0075425931	0.719544324	0.9989109	0.9999999	1.0000000
9	0.0000000024	0.0182884742	0.831310374	0.9997592	1.0000000	1.0000000
10	0.0000000147	0.0393399348	0.907695939	0.9999525	1.0000000	1.0000000
11	0.0000000803	0.0759167495	0.953997555	0.9999916	1.0000000	1.0000000
12	0.0000003921	0.1327163900	0.979081569	0.9999987	1.0000000	1.0000000
13	0.0000017253	0.2120472522	0.991303922	0.9999998	1.0000000	1.0000000
14	0.0000068805	0.3122252175	0.996688431	1.0000000	1.0000000	1.0000000
15	0.0000249816	0.4271039913	0.998842575	1.0000000	1.0000000	1.0000000
16	0.0000829116	0.5471766693	0.999628065	1.0000000	1.0000000	1.0000000
17	0.0002524168	0.6619208147	0.999889936	1.0000000	1.0000000	1.0000000
18	0.0007070640	0.7624352216	0.999969965	1.0000000	1.0000000	1.0000000
19	0.0018273288	0.8433226512	0.999992433	1.0000000	1.0000000	1.0000000
20	0.0043677297	0.9032284794	0.999998238	1.0000000	1.0000000	1.0000000
21	0.0096772356	0.9441194463	0.999999621	1.0000000	1.0000000	1.0000000
22	0.0199167486	0.9698742973	0.999999924	1.0000000	1.0000000	1.0000000

23	0.0381539994	0.9848554620	0.999999986	1.0000000	1.0000000	1.0000000
24	0.0681706184	0.9929084017	0.999999998	1.0000000	1.0000000	1.0000000
25	0.1138420882	0.9969100980	1.000000000	1.0000000	1.0000000	1.0000000
26	0.1780903712	0.9987486128	1.000000000	1.0000000	1.0000000	1.0000000
27	0.2616427619	0.9995294688	1.000000000	1.0000000	1.0000000	1.0000000
28	0.3620531308	0.9998359453	1.000000000	1.0000000	1.0000000	1.0000000
29	0.4734962150	0.9999470361	1.000000000	1.0000000	1.0000000	1.0000000
30	0.5876265363	0.9999841924	1.000000000	1.0000000	1.0000000	1.0000000
31	0.6953522100	0.9999956464	1.000000000	1.0000000	1.0000000	1.0000000
32	0.7889299858	0.9999988959	1.000000000	1.0000000	1.0000000	1.0000000
33	0.8636060054	0.9999997428	1.000000000	1.0000000	1.0000000	1.0000000
34	0.9182322119	0.9999999451	1.000000000	1.0000000	1.0000000	1.0000000
35	0.9547666858	0.9999999893	1.000000000	1.0000000	1.0000000	1.0000000
36	0.9770377768	0.9999999981	1.000000000	1.0000000	1.0000000	1.0000000
37	0.9893664881	0.9999999997	1.000000000	1.0000000	1.0000000	1.0000000
38	0.9955370869	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
39	0.9983148413	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
40	0.9994324156	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
41	0.9998312039	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
42	0.9999562256	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
43	0.9999902552	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
44	0.9999981757	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
45	0.9999997207	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
46	0.9999999664	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
47	0.9999999970	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
48	0.9999999998	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000

49	1.0000000000	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000
50	1.0000000000	1.0000000000	1.000000000	1.0000000	1.0000000	1.0000000


Dan Toomey
R
for Data
Science
Learn and explore the fundamentals of data science with R
Packt>

A caption