STT465ProblemSet#2

Sam Isken

September 25, 2019

STT 465 - Fall 2019

Homework 2 - Due 10/02/2019 (In Class)

Instruction: -When using R in any problem, copy the code and results onto your word document under that question number and add any required comments. You will lose points if I do not see your codes.

You should present a stapled document when multiple pages are used. The grader will not be held responsible for any loss of pages.

1. Exercise 3.1 on Page 227 (Textbook)

3.1

Sample Survey: Suppose we are going to sample 100 individuals from a county (of size much larger than 100) and ask each sampled person whether they support policy or not.

Let if person i in the sample supports the policy, and otherwise.

1. Assume are, conditional on , i.i.d binary random variable with expectation .

Write down the joint distribution of $ Pr(Y\_1 = Y\_1,…Y\_100=y\_{100}| ) $

in a compact form.

Also write down the form of

$Pr(\_{i=1}^{100}Y\_i = y | ) $

The Joint Distribution of $ Pr(Y\_1 = Y\_1,…Y\_100=y\_{100}| ) $ is given by:

This is because we assume $ Y\_1 , … Y\_n $ are, conditional on $ $, i.i.d binary random variable with expectation $ $.

For the second part of the question let us examine $ Pr(\_{i=1}^{100}Y\_i = y | ) $

1. For the moment, suppose you believed that ${0.0, 0.1, . ,0.9, 1.0} $. Given that the result of the survey were , compute $\_{i=1}^{100} Y\_i = 57 | ) $ for each of these 11 values of and plot these probabilities as a function of .

We first examine the Joint Distribution:

For this problem we start back at our joint distribution. We then will assume (given the context of the problem) that

for a set of ’s :

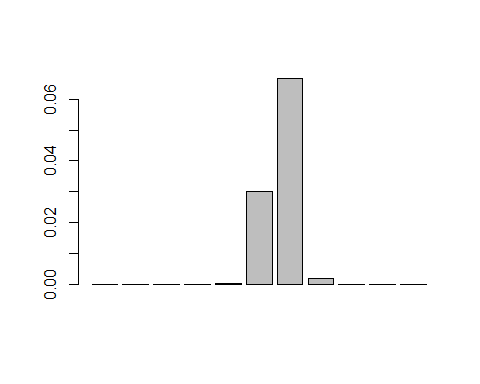
Thus:

Let us now implement this in R:

#Create set of thetas  
theta\_set = c(0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1)  
#Function to get values of joint pmf given set of thetas  
joint\_pmf <- function(set\_of\_theta){  
 for (variable in set\_of\_theta) {  
 theta <- variable  
 result <- (choose(100,57)) \* (theta^(57)) \* (1-theta)^(43)  
 print(result)  
 }  
}  
  
joint\_pmf(theta\_set)

## [1] 0  
## [1] 4.107157e-31  
## [1] 3.738459e-16  
## [1] 1.306895e-08  
## [1] 0.0002285792  
## [1] 0.03006864  
## [1] 0.06672895  
## [1] 0.001853172  
## [1] 1.003535e-07  
## [1] 9.395858e-18  
## [1] 0

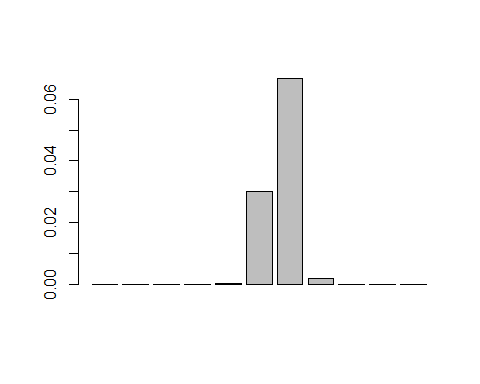
#List of values outputted from functino  
fin\_list <- c(0,4.107157e-31,3.738459e-16,1.306895e-08,0.0002285792,  
 0.03006864,0.06672895,0.001853172,1.003535e-07,9.395858e-18,0)  
  
#plot of posterior distribution  
posterior\_dist\_plot <- barplot(fin\_list)



1. Now suppose you originally has no prior information to believe one of these values over another, and so Pr( = 0.0) = Pr( = 0.1) = … = Pr( = 0.9) = Pr( = 1.0). Use Bayes’ rule to compute Pr() for each value. Make a plot of this posterior distribution as a function of .

The plot below shows our posterior distribition.

posterior\_dist\_plot <- barplot(fin\_list)



Now let us use Bayes’ rule to compute Pr() for each value.

Since:

print(length(theta\_set))

## [1] 11

We assume

And we know:

Posterior Distribution = Prior Distribution \* Likelihood Function

Thus:

$$ \text{Pr}(\theta|y) = \text(Posterior Distribution) = \frac{{100 \choose 57}\theta^{57} (1-\theta)^{43} (1/11)} {p(y)}$$

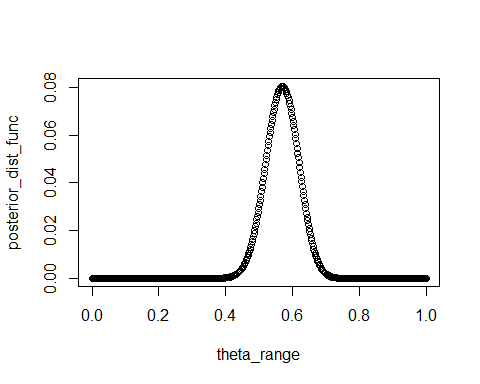
This is the desired answer, the posterior distribution.

1. Now suppose you allow to be any value in the interval [0,1]. Using the uniform prior density for , so that p() = 1, plot the posterior density

p() × Pr()

as a function of .

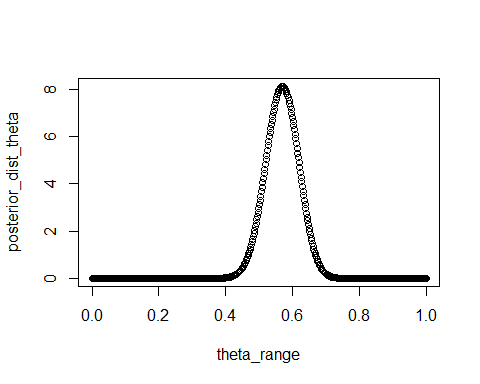
#Set theta to be the range [0,1]  
theta\_range <- seq(0,1,length=500)  
#Set theta probability  
prob\_theta = 1   
posterior\_dist\_func <- (choose(100,57))\*(theta\_range^57)\*(1-theta\_range)^43  
#Plot tjos posteriord distribution  
plot(theta\_range,posterior\_dist\_func)



1. As discussed in this chapter, the posterior distribution of is: beta(1 + 57, 1 + 100 - 57).

Plot the posterior density as a function of . Discuss the relationships among all of the plots you have made for this exercise.

posterior\_dist\_theta <- dbeta(theta\_range,1 + 57, 1 + 100 - 57)  
plot(theta\_range,posterior\_dist\_theta)



As you can see all of these graphs are equivalent. The posterior distribution of p() × Pr() when plotted is shown to be the same as ~beta(1 + 57, 1 + 100 - 57). This makes sense as we disscussed the flexibility of the Beta distribution and its similiarity in posteriors to other distributions.

1. The “fish” data set in homework 2 folder on D2L contains data on the # of fish caught by campers in a park.

#Use read csv function in order read in "fish.csv" data set   
fish <- read.csv(file = "fish.csv", header = TRUE)  
fish

## nofish livebait camper persons child xb zg count  
## 1 1 0 0 1 0 -0.896314561 3.050404787 0  
## 2 0 1 1 1 0 -0.558344960 1.746148944 0  
## 3 0 1 0 1 0 -0.401731014 0.279938877 0  
## 4 0 1 1 2 1 -0.956298113 -0.601525664 0  
## 5 0 1 0 1 0 0.436890960 0.527709126 1  
## 6 0 1 1 4 2 1.394485474 -0.707534790 0  
## 7 0 1 0 3 1 0.184716746 -3.398022175 0  
## 8 0 1 0 4 3 2.329106569 -5.450901508 0  
## 9 1 0 1 3 2 0.188386485 -1.527417779 0  
## 10 0 1 1 1 0 0.287689924 1.393890500 1  
## 11 0 1 0 4 1 1.990952730 -1.933189988 0  
## 12 0 1 1 3 2 1.317893147 -2.471574545 0  
## 13 1 0 0 3 0 0.298041672 1.591265202 1  
## 14 0 1 0 3 0 1.290873408 0.829534888 2  
## 15 0 1 1 1 0 -0.060889840 2.820579290 0  
## 16 1 1 1 1 0 0.370049208 2.158344984 1  
## 17 0 1 0 4 1 1.979093432 -3.069952726 0  
## 18 1 1 1 3 2 0.715337098 -1.952804923 0  
## 19 0 1 1 2 1 1.516053081 -0.186567351 1  
## 20 0 1 0 3 1 -0.034895968 -0.118922494 0  
## 21 0 1 0 4 1 1.178230286 0.001856566 1  
## 22 0 1 1 4 0 1.642111778 1.892821312 5  
## 23 0 1 1 2 1 0.597727358 -0.294278145 0  
## 24 0 1 1 2 0 1.139723063 1.931791067 3  
## 25 0 1 1 3 0 3.500297546 1.451270819 30  
## 26 0 1 1 2 0 -0.789978385 2.817448616 0  
## 27 0 1 1 4 0 2.662356138 1.656562567 13  
## 28 0 1 0 2 1 1.606172442 -1.064542532 0  
## 29 0 1 0 1 0 0.018550180 0.080798268 0  
## 30 0 1 0 4 3 3.034559011 -4.824044704 0  
## 31 0 1 0 1 0 0.051000755 0.921823859 0  
## 32 0 1 1 3 1 0.745258808 -0.663867116 0  
## 33 0 1 1 4 0 2.453136683 3.508367777 11  
## 34 0 1 1 4 1 2.352706671 0.176609725 5  
## 35 0 1 1 1 0 -1.108257532 0.772088408 0  
## 36 1 0 0 2 0 0.515419424 1.656657577 1  
## 37 1 1 1 2 1 1.982768536 -0.642237127 1  
## 38 1 1 1 4 1 2.066855907 1.244507432 7  
## 39 1 1 1 3 1 0.095011771 -2.268660784 0  
## 40 0 1 1 4 1 5.352674007 -1.472992539 14  
## 41 0 1 1 1 0 -0.711813927 3.020478010 0  
## 42 0 1 1 4 0 3.484046221 2.355652809 32  
## 43 1 0 0 3 2 2.400906324 -3.086473703 0  
## 44 1 0 1 4 0 0.376028478 2.676077843 1  
## 45 1 1 0 4 2 1.085056424 -2.654790401 0  
## 46 0 1 1 1 0 -1.067198753 2.133044958 0  
## 47 0 1 1 2 1 0.322997063 0.303436488 0  
## 48 0 1 0 1 0 0.501066327 1.553161621 1  
## 49 0 1 1 2 1 2.011488438 0.627036452 5  
## 50 0 1 0 2 1 0.957147360 -2.058174610 0  
## 51 0 1 1 2 1 1.114434004 0.284732074 1  
## 52 0 1 1 2 1 -0.673830032 -0.708101988 0  
## 53 0 1 1 4 0 3.196705818 1.515483379 22  
## 54 1 0 0 2 0 -0.386320800 2.079587936 0  
## 55 0 1 1 3 0 2.718511581 2.629312992 15  
## 56 1 0 1 1 0 -1.269290566 4.179599762 0  
## 57 0 1 1 1 0 -1.087805986 2.122612953 0  
## 58 0 1 1 1 0 -0.984578311 1.351520896 0  
## 59 1 1 1 4 1 1.872467756 1.261176825 5  
## 60 1 1 1 1 0 1.547955513 1.628988385 4  
## 61 0 1 0 2 0 1.004094243 1.083624601 2  
## 62 0 1 1 2 1 -0.165202618 2.095250368 0  
## 63 0 1 0 2 1 1.471645236 -0.073470898 2  
## 64 0 1 1 4 0 3.489336014 2.547997475 32  
## 65 0 1 0 4 3 1.885775805 -4.232519627 0  
## 66 0 1 1 1 0 -2.272530079 1.600753188 0  
## 67 0 1 0 1 0 0.613840997 1.111755967 1  
## 68 1 1 0 3 2 2.878997564 -2.766067028 0  
## 69 1 1 1 1 0 -0.944848835 2.011607885 0  
## 70 0 1 1 2 1 0.820035219 -1.285437942 0  
## 71 0 1 1 3 0 2.088183641 2.268748283 7  
## 72 0 1 0 4 3 2.165306330 -5.035178185 0  
## 73 1 0 0 4 2 0.557526350 -2.696651697 0  
## 74 0 1 1 3 2 -0.627695501 -3.224311590 0  
## 75 1 0 1 1 0 -3.275050163 0.913391829 0  
## 76 0 1 0 2 0 0.307397574 -0.431482762 0  
## 77 0 1 0 3 2 0.459303737 -3.140106678 0  
## 78 1 1 1 1 0 -0.188099161 3.267445326 0  
## 79 0 1 0 2 1 -0.197423220 -0.238331914 0  
## 80 0 1 0 4 0 0.901133239 1.393922210 2  
## 81 0 1 1 4 1 2.229737520 -0.410112113 3  
## 82 1 1 0 2 1 1.150078893 -0.320476443 1  
## 83 0 1 1 3 0 1.716530919 2.654059887 5  
## 84 0 1 0 1 0 -0.465738386 0.246082529 0  
## 85 0 1 1 1 0 1.019573331 1.718844175 2  
## 86 1 0 0 3 1 1.867413759 -0.548479140 1  
## 87 1 0 0 4 1 0.714378536 -2.550681114 0  
## 88 1 1 1 1 0 0.044008121 2.262954950 1  
## 89 0 1 1 4 0 5.005039692 3.572134256 149  
## 90 0 1 1 3 2 2.461556435 -2.769872427 0  
## 91 0 1 1 3 1 1.570416808 -0.390616268 1  
## 92 0 1 0 2 0 -1.490852475 0.088999458 0  
## 93 0 1 1 3 0 -0.865133345 0.972079754 0  
## 94 0 1 0 2 1 0.833803177 0.023357686 1  
## 95 0 1 0 4 2 2.208499908 -1.734373331 0  
## 96 0 1 1 3 1 1.633207917 -1.501252651 0  
## 97 0 1 0 4 2 1.524009705 -4.324279785 0  
## 98 1 0 0 4 0 2.491374731 -0.722057521 2  
## 99 0 1 1 2 0 1.084769011 2.963002682 2  
## 100 0 1 0 4 0 3.594677210 0.860208869 29  
## 101 0 1 1 1 0 1.128834605 2.060700417 3  
## 102 1 0 1 2 0 -0.385939926 2.266276360 0  
## 103 0 1 0 4 2 2.213509560 -2.068094730 0  
## 104 1 0 0 3 0 2.518778086 -0.110921405 5  
## 105 0 1 1 2 0 0.011589484 -0.339218020 0  
## 106 0 1 0 4 1 3.298806190 -2.671430111 0  
## 107 0 1 1 1 0 -0.258302957 0.714798093 0  
## 108 0 1 0 3 1 0.931161046 -1.925231814 0  
## 109 0 1 1 4 1 3.234340668 -1.934120536 0  
## 110 0 1 1 4 1 0.566562116 1.361151457 1  
## 111 0 1 1 1 0 1.974095106 3.178084135 7  
## 112 0 1 1 2 0 0.109922774 2.124498606 1  
## 113 0 1 0 1 0 -1.052196026 -1.837709427 0  
## 114 1 0 0 4 1 1.193210363 0.732807636 2  
## 115 0 1 0 3 2 0.351580232 -2.184268236 0  
## 116 1 1 0 2 0 1.180063128 1.293958902 2  
## 117 0 1 0 2 1 0.442351252 -0.252750695 0  
## 118 0 1 0 4 1 0.339507729 -0.768189728 0  
## 119 1 0 1 3 1 0.368057549 -0.848371208 0  
## 120 1 1 1 3 2 2.431215048 -0.945337653 1  
## 121 0 1 0 2 1 -0.267399877 -1.620429158 0  
## 122 1 1 0 1 0 -0.366596520 2.514541626 0  
## 123 1 0 1 4 3 1.481658340 -2.820960760 0  
## 124 0 1 0 1 0 -0.620765746 0.583114088 0  
## 125 1 0 1 4 3 1.593668580 -2.947399616 0  
## 126 0 1 0 4 1 2.824294329 -0.742032945 3  
## 127 0 1 1 1 0 1.487538576 2.265225410 4  
## 128 0 1 1 3 0 1.139073014 3.448698759 3  
## 129 0 1 0 2 0 1.415821791 1.297202468 3  
## 130 0 1 1 4 0 2.172077894 4.263185024 8  
## 131 1 0 1 3 0 0.818082452 3.058596849 2  
## 132 1 1 1 1 0 0.332590669 2.045859098 1  
## 133 0 1 0 4 0 1.856842756 2.516795874 6  
## 134 1 1 1 4 2 0.593688667 -1.977207899 0  
## 135 0 1 1 2 1 0.061764006 1.269807458 0  
## 136 1 1 0 4 0 1.984782338 0.824614167 5  
## 137 0 1 1 4 1 1.963856339 0.031520370 3  
## 138 0 1 0 3 1 3.726072550 0.701675057 31  
## 139 0 1 1 2 1 0.349324912 -0.161479443 0  
## 140 0 1 0 2 0 0.704055548 2.344917059 2  
## 141 1 1 1 4 3 1.569243789 -4.469278812 0  
## 142 0 1 1 2 1 0.781064868 -1.890136719 0  
## 143 1 0 0 3 0 0.139555037 0.339156687 0  
## 144 0 1 1 3 1 1.541620493 -1.531070232 0  
## 145 1 1 0 1 0 0.071873553 1.305795789 0  
## 146 0 1 0 2 0 0.746514857 -0.650594831 0  
## 147 1 1 1 4 0 1.925082445 3.567077875 6  
## 148 0 1 0 3 0 2.214212179 2.691571951 9  
## 149 1 0 1 3 2 -0.599701524 -2.059751987 0  
## 150 1 0 1 2 1 -2.107331038 0.141346171 0  
## 151 0 1 1 1 0 -2.490455389 2.073759556 0  
## 152 0 1 1 2 1 -0.070476264 -0.478490353 0  
## 153 0 1 0 1 0 -0.233503744 0.152506366 0  
## 154 0 1 0 2 0 2.148858786 -0.488574415 2  
## 155 0 1 0 3 0 3.173754692 0.378390342 15  
## 156 0 1 1 1 0 0.676705837 1.623785019 1  
## 157 1 1 1 2 0 0.903368711 2.015382290 2  
## 158 0 1 1 3 1 2.121471882 -0.091606267 3  
## 159 1 1 1 1 0 -1.680656552 2.277235031 0  
## 160 0 1 1 4 0 4.268487930 1.362295985 65  
## 161 0 1 1 3 0 1.759671211 1.478169680 5  
## 162 0 1 0 1 0 0.007159876 0.184564099 0  
## 163 1 1 1 3 2 3.533810616 -3.252695322 0  
## 164 0 1 0 4 1 1.902828455 -1.590401888 0  
## 165 0 1 0 4 2 2.054314375 -3.139189243 0  
## 166 1 1 0 1 0 0.132048607 2.244067192 1  
## 167 0 1 1 4 1 2.844588757 -0.078207836 8  
## 168 0 1 1 3 0 -0.150227517 2.496215105 0  
## 169 0 1 1 3 2 0.377593964 -2.294372320 0  
## 170 0 1 1 4 3 1.461001873 -3.943372011 0  
## 171 0 1 1 3 0 0.951952934 2.606308937 2  
## 172 0 1 1 2 1 1.531072855 1.537260294 4  
## 173 1 1 0 4 0 1.716095209 1.826901793 5  
## 174 0 1 0 2 0 2.242041826 2.398617029 9  
## 175 0 1 0 4 2 3.389037609 -2.893302679 0  
## 176 0 1 0 2 0 -0.457464665 3.179333210 0  
## 177 0 1 1 2 0 -0.263669461 1.760868073 0  
## 178 0 1 0 1 0 -0.712437570 1.177165627 0  
## 179 1 1 1 2 0 3.053916931 2.469107866 21  
## 180 0 1 0 1 0 -2.708276749 0.347330987 0  
## 181 0 1 1 3 0 1.872051477 2.307969570 6  
## 182 0 1 0 2 1 0.169749007 -0.835122049 0  
## 183 1 0 1 2 0 -0.348856091 2.941457510 0  
## 184 0 1 1 2 1 0.240219221 -0.403173774 0  
## 185 1 1 0 3 1 2.403877020 -1.568528771 0  
## 186 0 1 1 4 1 2.988574505 0.950103223 16  
## 187 1 0 0 3 2 0.005257762 -3.463099957 0  
## 188 0 1 1 3 2 0.400785357 -1.719052434 0  
## 189 1 1 1 4 0 1.592312098 3.472146988 4  
## 190 0 1 1 2 1 1.022531271 0.598669052 2  
## 191 0 1 1 3 0 2.377341747 2.578616858 10  
## 192 1 0 1 1 0 -2.112927198 1.630112410 0  
## 193 0 1 1 1 0 -0.557507157 1.938125849 0  
## 194 1 0 0 1 0 -1.276221037 0.145860225 0  
## 195 1 0 1 2 0 1.060786009 2.973175526 2  
## 196 1 1 1 2 0 0.409971446 1.287543416 1  
## 197 0 1 0 1 0 1.472747684 1.001728415 3  
## 198 0 1 1 3 1 1.122244954 -1.014991045 0  
## 199 0 1 1 4 2 2.242376328 -2.386286736 0  
## 200 0 1 1 2 0 3.163843393 1.296834946 21  
## 201 0 1 0 2 1 -0.585306764 -2.078762770 0  
## 202 0 1 0 2 0 -0.015607119 2.299016714 0  
## 203 1 0 1 1 0 0.834488332 2.205511093 2  
## 204 1 0 1 2 1 -1.268876076 1.570919514 0  
## 205 0 1 1 4 1 2.757081985 -0.866619527 3  
## 206 0 1 1 3 1 1.703949213 -1.154581547 0  
## 207 0 1 1 4 0 3.765002489 1.189270258 38  
## 208 1 1 1 4 3 2.719051123 -4.614192963 0  
## 209 0 1 1 1 0 -0.154021844 2.465320110 0  
## 210 0 1 0 1 0 -0.411525637 0.257135123 0  
## 211 0 1 1 1 0 0.531122029 2.928965569 1  
## 212 0 1 0 1 0 1.340306163 3.566372156 3  
## 213 0 1 1 1 0 -1.562050104 1.406773925 0  
## 214 0 1 0 2 0 0.225383952 2.220991135 1  
## 215 1 1 0 4 2 4.502663612 -4.730084896 0  
## 216 1 0 0 2 1 0.162199765 -1.125854254 0  
## 217 0 1 0 4 2 3.047013521 -3.184262991 0  
## 218 0 1 0 3 1 1.630753398 -2.707653284 0  
## 219 0 1 1 4 2 1.983255029 0.573806643 5  
## 220 0 1 0 4 2 3.877111435 -2.179812431 0  
## 221 1 0 0 4 1 1.584755540 -1.289599061 0  
## 222 0 1 0 3 0 1.701143265 0.005117925 2  
## 223 1 1 1 2 1 -0.170604989 1.469863534 0  
## 224 0 1 1 1 0 -0.496980190 0.957906127 0  
## 225 0 1 0 2 1 1.218269944 -0.664241612 0  
## 226 0 1 1 2 0 0.037536614 2.861773014 1  
## 227 1 1 1 3 1 1.523244143 3.039572716 4  
## 228 1 1 0 1 0 -0.907698154 0.611488819 0  
## 229 0 1 1 3 2 2.331376553 -2.355525732 0  
## 230 0 1 1 2 1 1.068086028 1.375056386 2  
## 231 0 1 1 1 0 1.401265740 2.141049623 3  
## 232 1 1 0 4 2 3.930421829 -2.743903160 0  
## 233 0 1 0 3 1 0.196246982 -1.781073928 0  
## 234 0 1 0 2 0 0.029527381 1.002204299 0  
## 235 1 0 0 2 0 -0.766901433 -0.442087710 0  
## 236 0 1 1 1 0 0.661188662 3.241333485 1  
## 237 0 1 1 2 0 1.015074015 0.948959351 2  
## 238 0 1 1 1 0 -1.046668053 0.760976613 0  
## 239 0 1 1 3 0 1.938054800 2.159034252 6  
## 240 0 1 1 2 0 1.724750519 0.927846849 4  
## 241 0 1 1 2 0 0.602655232 3.571609020 1  
## 242 0 1 1 4 2 2.413781643 -1.316399097 1  
## 243 1 0 0 2 0 -1.200768590 1.057799816 0  
## 244 0 1 1 3 1 1.963849306 -0.733001232 1  
## 245 0 1 0 3 0 -0.291065693 1.315508246 0  
## 246 1 1 1 2 0 -0.755235732 2.324208736 0  
## 247 0 1 1 4 3 1.794859171 -5.625943661 0  
## 248 0 1 1 2 1 -0.392648846 0.677275419 0  
## 249 1 1 1 3 2 1.374640584 -2.595630169 0  
## 250 1 1 1 2 1 0.828834116 -1.457115412 0

1. Present a frequency table using R with observed frequencies for x (x=# of fish caught)

table(fish$count)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 13 14 15 16 21 22   
## 142 31 20 12 6 10 4 3 2 2 1 1 1 1 2 1 2 1   
## 29 30 31 32 38 65 149   
## 1 1 1 2 1 1 1

1. Assuming a Poisson model, provide the maximum likelihood estimate of the Poisson parameter (“lambda”) and an approximate 95% CI.

The PDF of the poisson distribution is given by.

To find the Maximum likelihood estimate of we must have (for iid Poisson RVs) a joint frequency that is the product of the marginal frequency functions, Therefore, the log likelihood is given by:

If we simplify further it is clear that:

We know (from STT 442 - Mathematical Statistics) that we find the MLE for an estimator by taking the first derivitive of a likelihood function and setting it equal to 0:

This implies that the estimate of the estimator in the Poisson distribution is given by:

Since, under the Poisson distribution:

This implies the 95% CI is given by:

Thus in our case:

#Frequency table   
table(fish$count)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 13 14 15 16 21 22   
## 142 31 20 12 6 10 4 3 2 2 1 1 1 1 2 1 2 1   
## 29 30 31 32 38 65 149   
## 1 1 1 2 1 1 1

#Find our lambda hat   
lambda\_hat <- mean(table(fish$count))  
print(lambda\_hat)

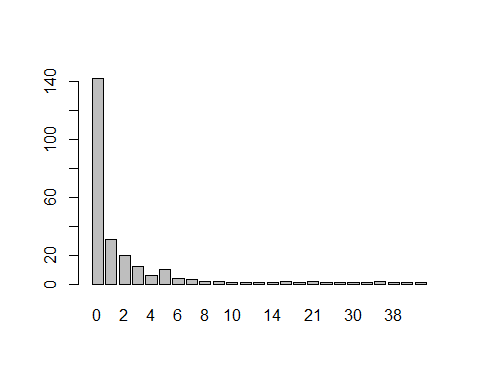
## [1] 10

#Get length of table, n for CI  
n <- length(table(fish$count))  
CI = c(lambda\_hat-(1.96\*sqrt(lambda\_hat/n)),lambda\_hat+(1.96\*sqrt(lambda\_hat/n)))  
print(CI)

## [1] 8.760387 11.239613

1. Present in a bar-plot the observed frequencies (from question a) and the predicted frequencies according to a Poisson model with a rate parameter equal to the maximum likelihood estimate that you reported in (b).

#Bar plot of the observed frequencies   
barplot(table(fish$count))



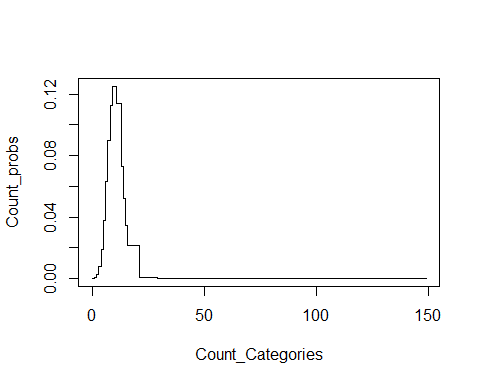
#Create vector of categories tp predict   
Count\_Categories <- c(0,1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,21,22,29,30,31,32,38,65,149)  
lambda\_hat

## [1] 10

#Create e   
e <- exp(1)  
  
poisson\_func <- function(Set\_of\_counts){  
 for (variable in Set\_of\_counts) {  
 x <- variable  
 result <- (e^(-10) \* 10^x)/factorial(x)  
 print(result)  
 }  
}  
  
poisson\_func(Count\_Categories)

## [1] 4.539993e-05  
## [1] 0.0004539993  
## [1] 0.002269996  
## [1] 0.007566655  
## [1] 0.01891664  
## [1] 0.03783327  
## [1] 0.06305546  
## [1] 0.09007923  
## [1] 0.112599  
## [1] 0.12511  
## [1] 0.12511  
## [1] 0.1137364  
## [1] 0.07290795  
## [1] 0.0520771  
## [1] 0.03471807  
## [1] 0.02169879  
## [1] 0.0008886101  
## [1] 0.0004039137  
## [1] 5.134715e-07  
## [1] 1.711572e-07  
## [1] 5.521199e-08  
## [1] 1.725375e-08  
## [1] 8.6803e-12  
## [1] 5.504589e-31  
## [1] 1.191936e-116

#Out put of function  
Count\_probs <- c(4.539993e-05,0.0004539993,0.002269996,0.007566655,0.01891664,0.03783327,0.06305546,0.09007923,0.112599,0.12511,0.12511,0.1137364,0.07290795,0.0520771,0.03471807,0.02169879,0.0008886101,0.0004039137,5.134715e-07,1.711572e-07,5.521199e-08,1.725375e-08,8.6803e-12,5.504589e-31,1.191936e-116)  
  
#Plotting predicted values to compare   
plot(Count\_Categories,Count\_probs,type = "s")



I plotted the predicted probability of each count above in order to compare with the actual values.

1. Comment the results that you obtained in c. Does the Poisson model fits well this data?

From this we can tell the Poisson model does fit our data well! The plot of our predicted probabilities is relatively similiar to the plot of the actual occurences of each quantity. This is not suprising as our data was right skewed and the poisson distribution is often used to measure frequencies.

3.6

Exponential family expectation:

Let

be an exponential family model.

1. Take derivatives with respect to ???? of both sides of the equation

to show that

We begin with:

Let us now integrate:

Thus:

Thus for the exponential distribution:

Therefore

as desired.

1. Let

be the prior distribution for . Calculate and, using the fundemental theorem of calculus disscuss what must be true so that

We know that

If we integrate with resoect to $ $ we will get:

So

must be true to assume

3.9 Galenshore distribution: An unknown quantity Y has a Galenshore() distribution if its’ density is given by:

For y>0, >0 and >0. Assume is known, For this density:

1. Identify a class of conjugate prior densities for .

Let us denote a class of conjugate prior densistes for by:

1. Let .

Find the posterior distribution of given , using a prior from your conjugate class.

which can be further simplified to:

This change in layout shows that the posterior distribution of $ $ is: galenshore($ na +a\_0 , $).

1. Write down and simplify. Identify a sufficient statistic.

We can simplify this probability to the following:

From this we see that | y up to $*{i=1}^n y\_i^2 $. This implies that $*{i=1}^n y\_i^2 $ is a sufficient statistic for .

1. Determine