Problem Set 4

Adam Ten Hoeve

Introduction

We discussed parameter fitting in class, and saw examples of modeling data with a model from a parametrized family. In these examples the model with the optimal parameters fit the corresponding data fairly well. This depends on the model family's being well suited to the data. If it isn't, even the best parameters won't produce a model that closely reflects the data. Please be aware of this possibility as you work through the examples here.

Please complete the following tasks regarding the data in R. Please generate a solution document in R markdown and upload the .Rmd document and a rendered .doc, .docx, or .pdf document. Your work should be based on the data's being in the same folder as the .Rmd file. Please turn in your work on Canvas. Your solution document should have your answers to the questions and should display the requested plots. Each part is worth 5 points.

These questions were rendered in R markdown through RStudio (https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf, http://rmarkdown.rstudio.com).

Question 1

This question uses 2018 data primarily for Denver county accessed through IPUMS-USA, University of Minnesota, www.ipums.org ,

Steven Ruggles, Sarah Flood, Ronald Goeken, Josiah Grover, Erin Meyer, Jose Pacas and Matthew Sobek. IPUMS USA: Version 9.0 [dataset]. Minneapolis, MN: IPUMS, 2019. https://doi.org/10.18128/D010.V9.0

The PUMA-to-county restriction was done using MABLE, http://mcdc.missouri.edu/websas/geocorr12.html

IPUMS Data

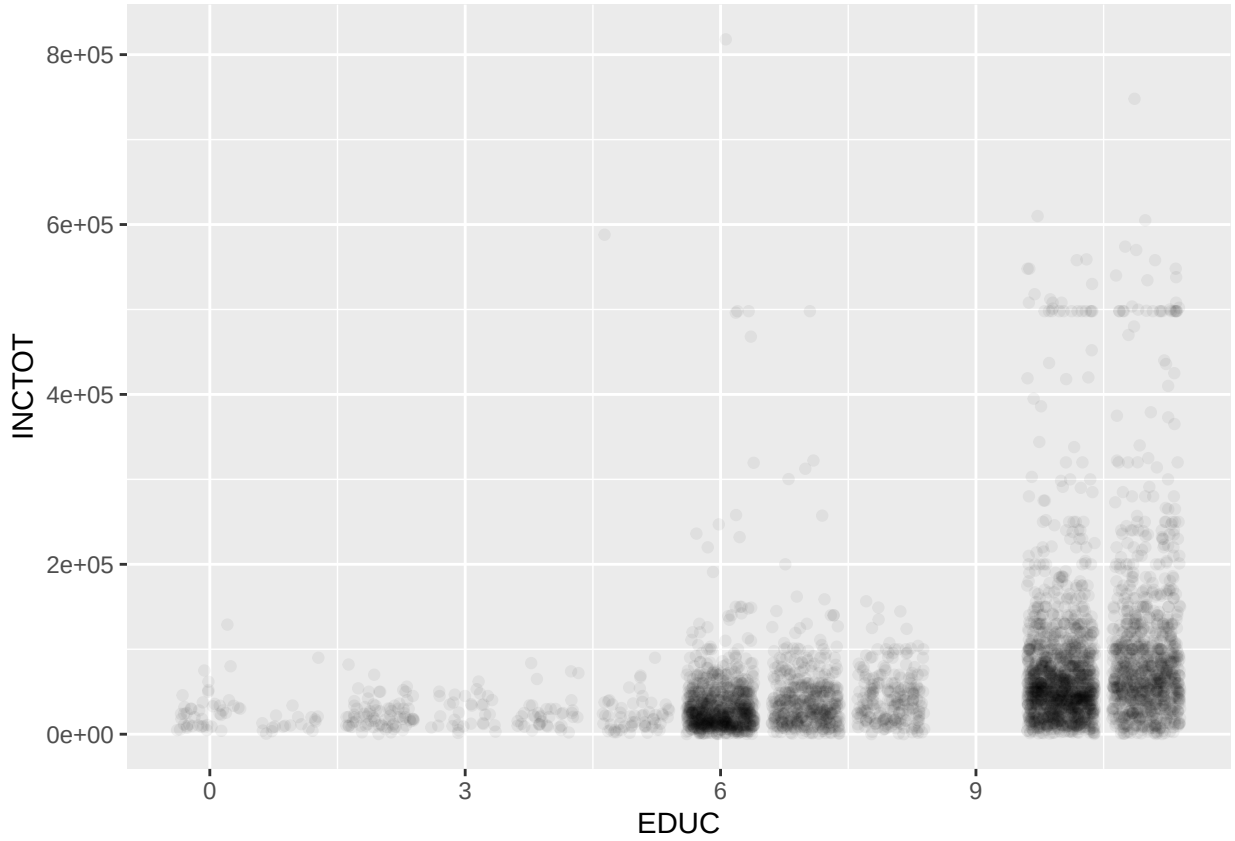
Read in the ipums data

```
dat<-read.csv("usa_00016_trim.csv")
```

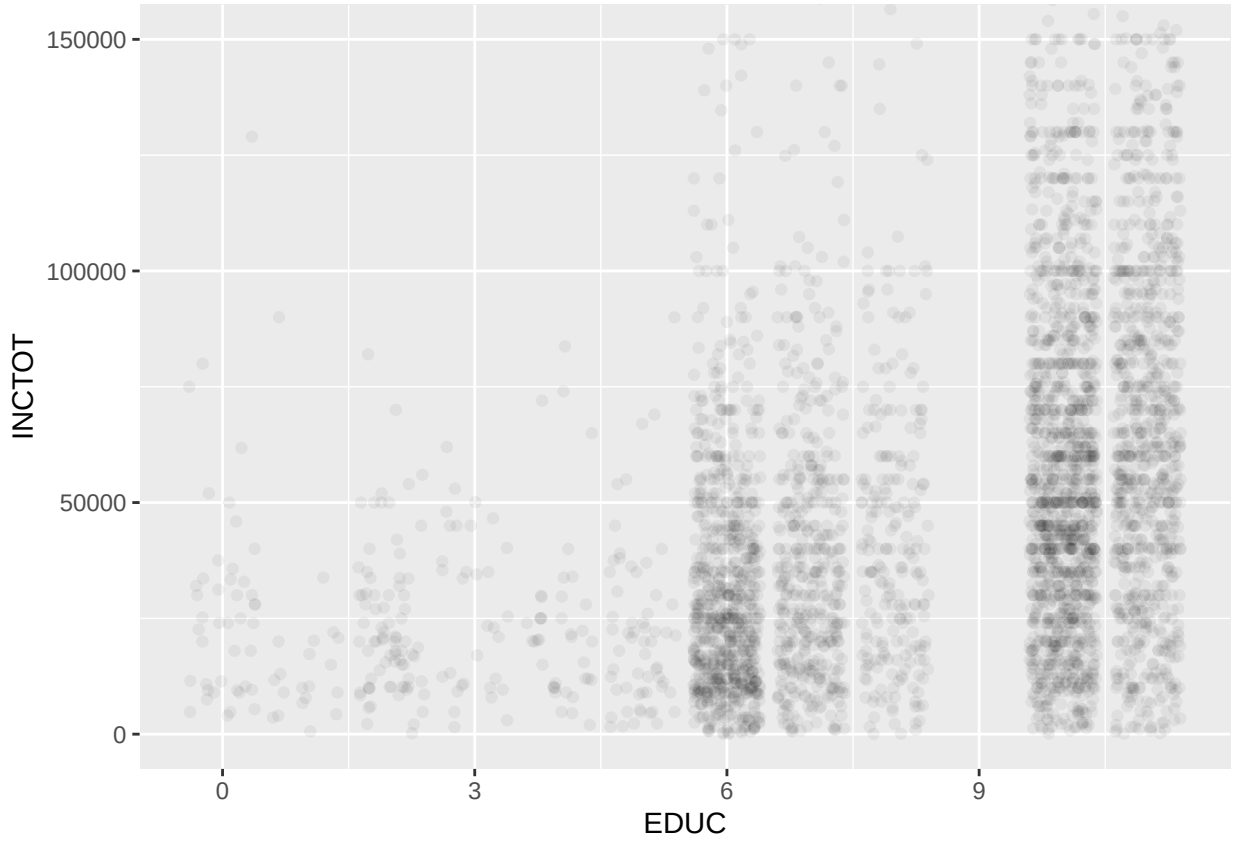
Trim to PUMAs that are predominantly in Denver. Restrict to respondents who are at least 25 years old with non-zero income. The variable "INCTOT" gives total annual individual income. The variable "EDUC" consists of ordered categories of amount of formal education, with 0 representing the least and 11 representing the most. Details are in the "educ_codes.csv".

The plot shows the "jitter" tool and the "alpha" approach for displaying large numbers of data points without losing information through plotting points on top of each other.

```
denver<-filter(dat,PUMA>=812 & PUMA<=816, INCTOT>0,AGE>=25)
g<-ggplot(denver,aes(x=EDUC,y=INCTOT))+geom_jitter(alpha=.05)
g
```




```
g<-g+coord_cartesian(ylim =c(0,1.5e5))
g
```



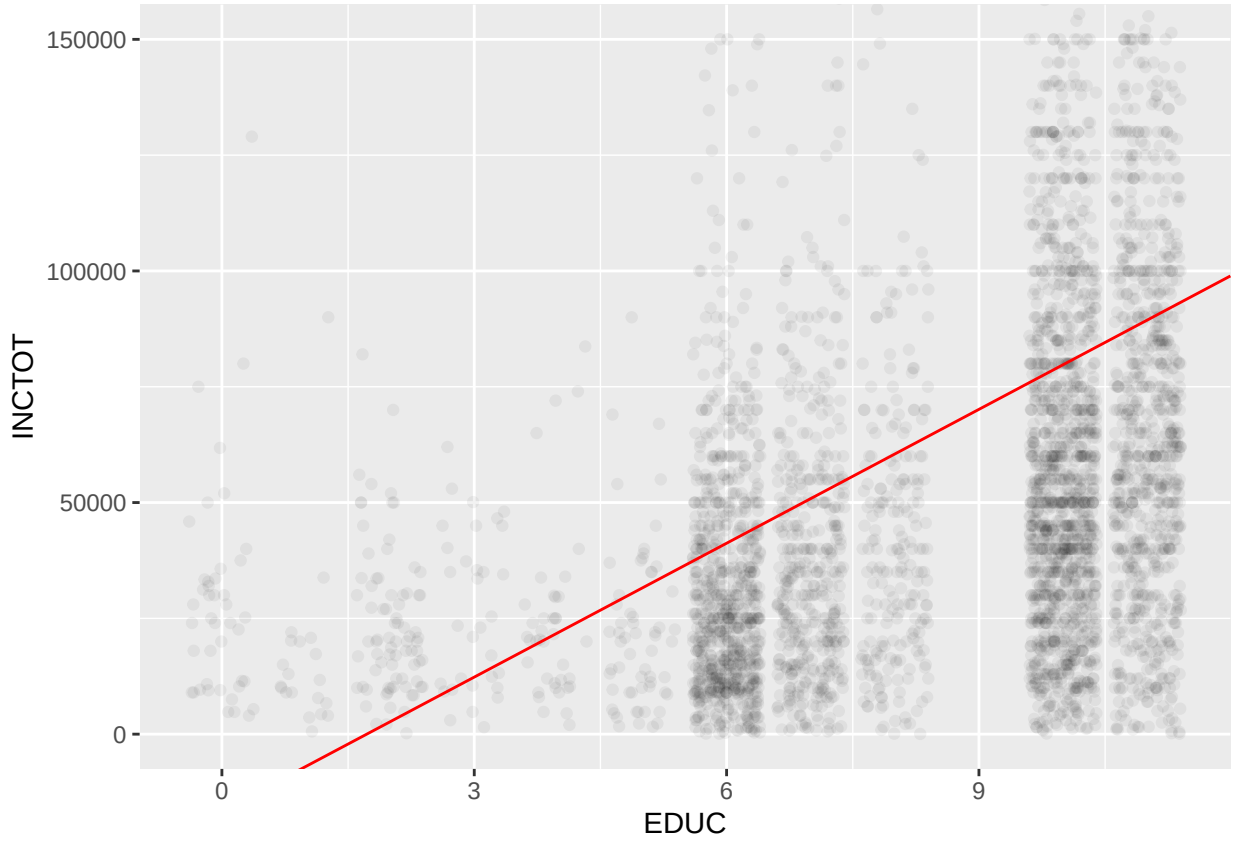
Question 1.a (5 points)

Please add the least squares best fit line in orange to the plot "g" above, save the result as "g", and display the result. What change in income is associated with an increase of 1 in the "EDUC" category?

```
model <- lm(INCTOT~EDUC, data=denver)
model$coef
```

```
## (Intercept)        EDUC
##  -16587.367    9634.065
```

```
g <- g + geom_abline(intercept=model$coef[1], slope=model$coef[2], color="red")
g
```



The slope of the regression line is 9634.065. That means that for every 1 increases in EDUC, the income increases by 9634.065.

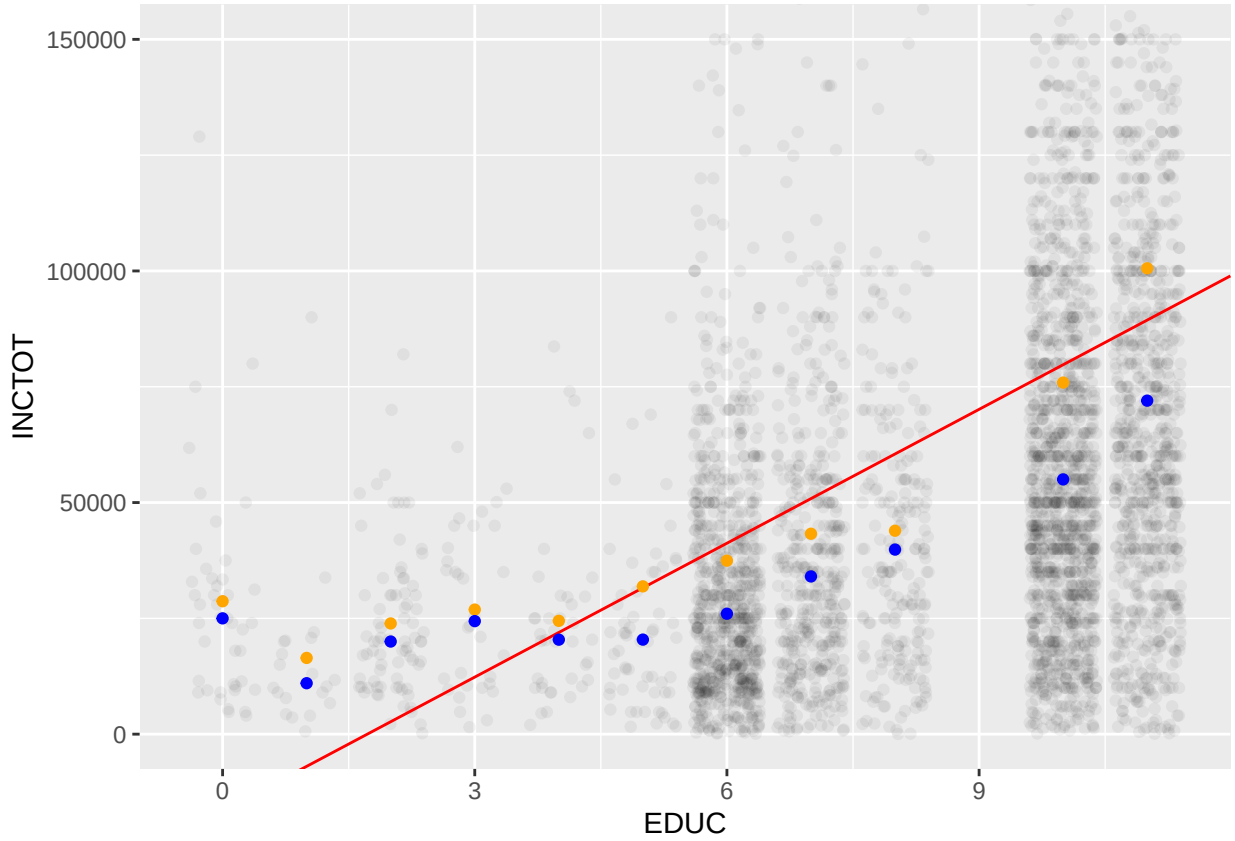
Question 1.b (5 points)

Please add the income means for each education category in orange and the income medians (R function "median") in each category in blue to the plot "g", save the result as "g", and display the result.

```
denver.summary = denver %>% group_by(EDUC) %>% summarize(mean = mean(INCTOT), median=median(INCTOT), n=
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
g <- g + geom_point(data=denver.summary, mapping=aes(x=EDUC, y=mean), color="orange") +
     geom_point(aes(x=EDUC, y=median), denver.summary, color="blue")
g
```



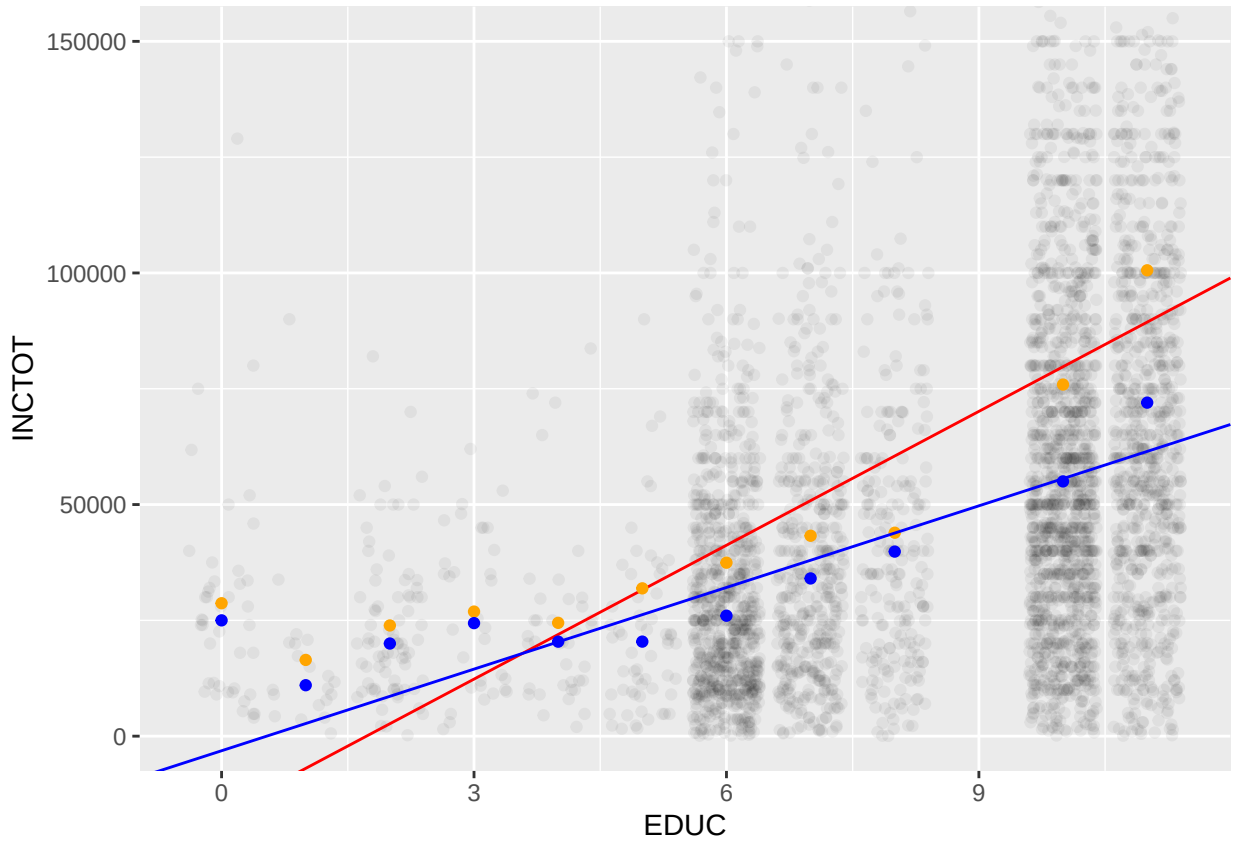
Question 1.c (5 points)

The least squares criterion is one way of fitting a line to a collection of points $\{(x_1, y_2), (x_2, y_2), ...(x_n, y_n)\}$.

One alternative is to fit the line that minimizes the sum of the absolute errors $|y_i - (mx_i + b)|$.

Please use "nlm" to fit this line for "INCTOT" as a function of "EDUC" and add this line in blue to "g", save the result as "g", and display the result. The slope and intercept from 1.a are option for the starting parameters. Other starting parameters may give other lines.

```
# Define the objective function
f <- function(theta, x, y){
  return (sum(abs(y - theta[1] - theta[2]*x)))
}

# Find the estimates using nlm, with the starting estimates
# being the intercept and slope from 1.a
estimates <- nlm(f, c(model$coef[1], model$coef[2]), x=denver$EDUC, y=denver$INCTOT)
g <- g + geom_abline(intercept=estimates$estimate[1], slope=estimates$estimate[2], color="blue")
g
```



Question 1.d (5 points)

Please redo 1.a-1.c restricting first to "EDUC" less than or equal to 5, then to "EDUC" greater than or equal to 5.

Here's the case for when EDUC is less than or equal to 5.

```
denver.le5<-filter(dat,PUMA>=812 & PUMA<=816, INCTOT>0,AGE>=25, EDUC<=5)
g2 <- ggplot(denver.le5,aes(x=EDUC,y=INCTOT))+geom_jitter(alpha=.20)
g2 <- g2 + coord_cartesian(ylim =c(0,1.5e5))
model.le5 <- lm(INCTOT~EDUC, data=denver.le5)
model.le5$coef
```

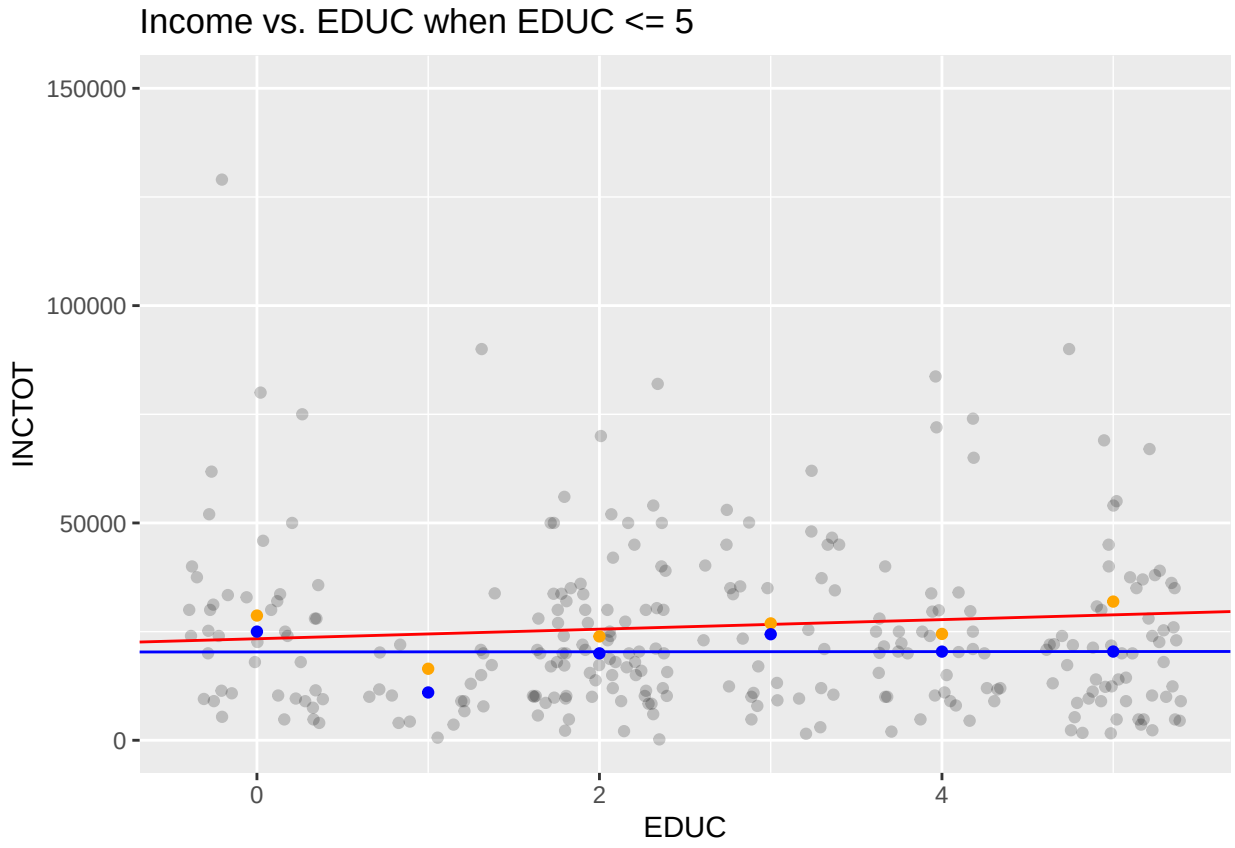
```
## (Intercept)        EDUC
##   23357.868    1100.592
```

```
g2 <- g2 + geom_abline(intercept=model.le5$coef[1], slope=model.le5$coef[2], color="red")
denver.summary.le5 = denver.le5 %>% group_by(EDUC) %>% summarize(mean = mean(INCTOT), median=median(INC
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
g2 <- g2 + geom_point(data=denver.summary.le5, mapping=aes(x=EDUC, y=mean), color="orange") +
           geom_point(aes(x=EDUC, y=median), denver.summary.le5, color="blue")
estimates.le5 <- nlm(f, c(model.le5$coef[1], model.le5$coef[2]), x=denver.le5$EDUC, y=denver.le5$INCTOT
```

```
g2 <- g2 + geom_abline(intercept=estimates.le5$estimate[1], slope=estimates.le5$estimate[2], color="blue")
g2 + labs(title="Income vs. EDUC when EDUC <= 5")
```



When EDUC is restricted to less than or equal to 5, we get the above plots. From the slope of the regression, we can see that a 1 unit increase in EDUC would lead to a 1100.592 increase in income. Now let's look at the case when EDUC is greater than or equal to 5.

```
denver.ge5<-filter(dat,PUMA>=812 & PUMA<=816, INCTOT>0,AGE>=25, EDUC>=5)
g3 <- ggplot(denver.ge5,aes(x=EDUC,y=INCTOT))+geom_jitter(alpha=.20)
g3 <- g3 + coord_cartesian(ylim =c(0,1.5e5))
model.ge5 <- lm(INCTOT~EDUC, data=denver.ge5)
model.ge5$coef
```

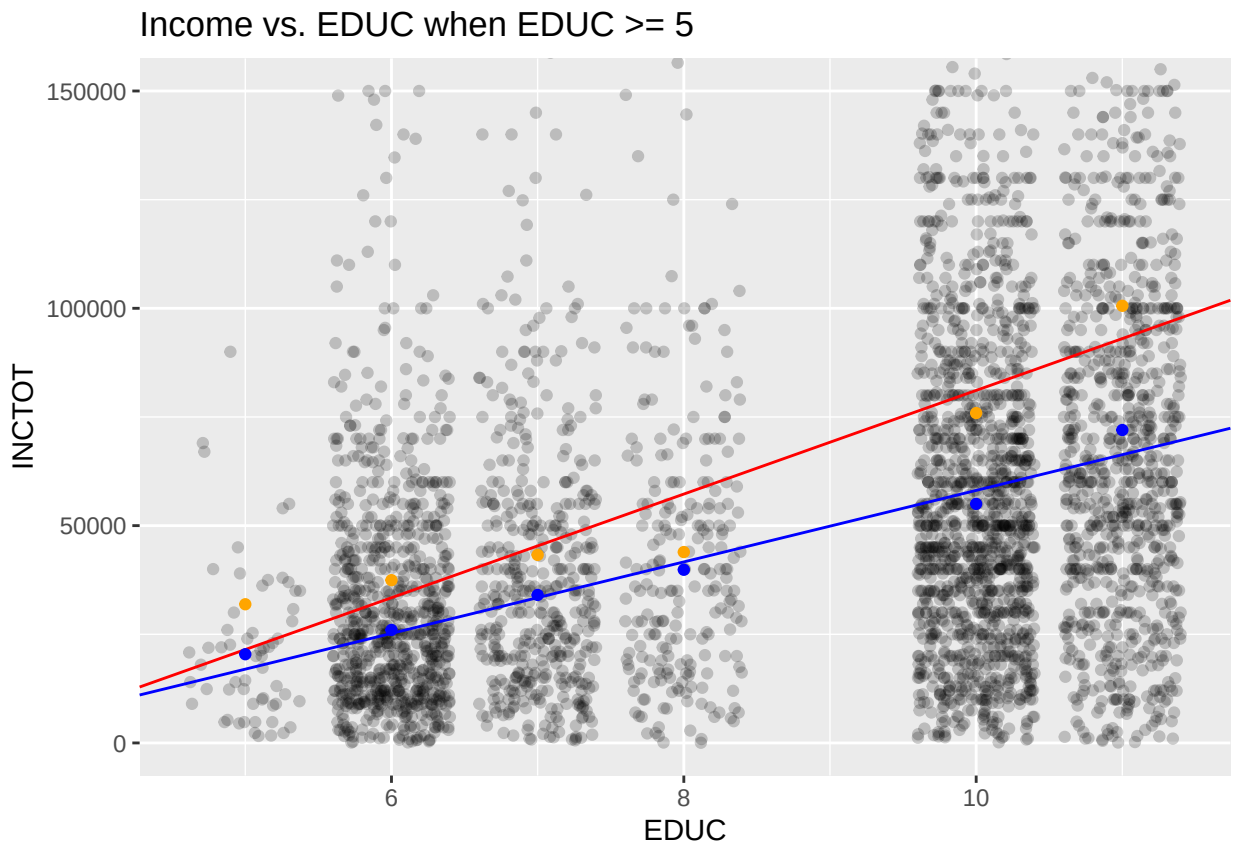
```
## (Intercept)        EDUC
##   -38131.94    11924.01
```

```
g3 <- g3 + geom_abline(intercept=model.ge5$coef[1], slope=model.ge5$coef[2], color="red")
denver.summary.ge5 = denver.ge5 %>% group_by(EDUC) %>% summarize(mean = mean(INCTOT), median=median(INC
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
g3 <- g3 + geom_point(data=denver.summary.ge5, mapping=aes(x=EDUC, y=mean), color="orange") +
           geom_point(aes(x=EDUC, y=median), denver.summary.ge5, color="blue")
estimates.ge5 <- nlm(f, c(model.ge5$coef[1], model.ge5$coef[2]), x=denver.ge5$EDUC, y=denver.ge5$INCTOT
```

```
g3 <- g3 + geom_abline(intercept=estimates.ge5$estimate[1], slope=estimates.ge5$estimate[2], color="blu
g3 + labs(title="Income vs. EDUC when EDUC >= 5")
```



This is the plot for when $EDUC >= 5$. We can see from the slope of the regression that if $EDUC$ changes by 1, then income will increase by $11,924.01$.

Question 1.e (5 points)

Please comment on the quality of the models fitted in parts 1.a-1.d. In particular, please identify the cases in which a line appears to be an appropriate summary of the data, identify the cases in which a line does not appear to be an appropriate summary of the data, and explain your reasoning. Also, please comment on the apparent relationship between the two types of fitted line and the two statistics, mean and median, for location of center. Finally, please comment on the size of the estimated change in "INCTOT" associated with a change in "EDUC" relative to the sample standard deviations of "INCTOT" within "EDUC" category.

When $EDUC <= 5$, a line is a good approximation of the data because there is not a lot of variation within each $EDUC$ category. For the other two cases, a line is a worse approximation because there is a greater spread among the data for each category.

From all three of the plots, we can see that the line fitted with the sum squared errors is closer to the means of each category. The line fit from the sum of absolute errors seems to follow the medians rather than the means.

We can see the difference in estimated change by looking at the plots when $EDUC <= 5$ to when $EDUC >= 5$. The first plot has much smaller sample standard deviations per category, and has a much smaller slope than the second plot, which has much larger sample standard deviations per category. Therefor we can say

that the associated change in $INCTOT$ is increased as the sample standard deviation per $EDUC$ category increases.

Question 2

This question uses the distribution means and variances to compare Poisson distributions and binomial distributions to corresponding Normal distributions.

Question 2.a (5 points)

The Poisson distribution with parameter λ is a discrete probability distribution with non-negative integer outcomes. The probability of the outcome k equals $\frac{\lambda^k}{k!}\exp(-\lambda)$. Given the following computations, identify the mean and variance of the Poisson distribution with parameter λ.

$$\sum_{k=0}^{\infty}\frac{\lambda^k}{k!} = \exp(\lambda)$$

and

$$\sum_{k=0}^{\infty}\frac{k\lambda^k}{k!} = \sum_{k=1}^{\infty}\frac{k\lambda^k}{k!}$$
$$= \lambda\sum_{k=1}^{\infty}\frac{\lambda^{k-1}}{(k-1)!} = \lambda\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}$$

and finally

$$\sum_{k=0}^{\infty}\frac{k^2\lambda^k}{k!} = \sum_{k=2}^{\infty}\frac{k(k-1)\lambda^k}{k!} + \sum_{k=0}^{\infty}\frac{k\lambda^k}{k!}$$
$$= \lambda^2\sum_{k=2}^{\infty}\frac{k(k-1)\lambda^{k-2}}{k!} + \lambda\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}$$
$$= \lambda^2\sum_{k=0}^{\infty}\frac{\lambda^k}{k!} + \lambda\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}$$

Let's start by solving for the mean, which will be the expectation of $X \sim Poisson(\lambda)$.

$$
\begin{aligned}
E[X] = \sum_{k \in S} k f(k) &= \sum_{k=0}^{n} k(e^{-\lambda}\frac{\lambda^k}{k!}) \\
&= \sum_{k=1}^{\infty} e^{-\lambda}\frac{k\lambda^k}{k!} \\
&= \sum_{k=1}^{\infty} e^{-\lambda}\frac{\lambda^k}{(k-1)!} \\
&= \sum_{k=1}^{\infty} e^{-\lambda}\frac{(\lambda)\lambda^{k-1}}{(k-1)!} \\
&= \lambda\sum_{k=1}^{\infty} e^{-\lambda}\frac{\lambda^{k-1}}{(k-1)!} \\
&= \lambda\sum_{k=0}^{\infty} e^{-\lambda}\frac{\lambda^k}{k!}
\end{aligned}
$$

Note that $e^{-\lambda}\frac{\lambda^k}{k!}$ is the PMF for a Poisson random variable at X=x, so $\sum_{k=0}^{\infty} e^{-\lambda}\frac{\lambda^k}{k!} = 1$. Thefore our final answer is:

$$\boxed{E[X] = \lambda}$$

Now to solve for the variance, we want to use the equation $Var(X) = E[X^2] - E[X]^2$ so we will need to solve for $E[X^2]$.

$$
\begin{aligned}
E[X^2] = \sum_{k \in S} k^2 f(k) &= \sum_{k=0}^{\infty} k^2(e^{-\lambda}\frac{\lambda^k}{k!}) \\
&= \sum_{k=0}^{\infty} e^{-\lambda}\frac{k^2\lambda^k}{k!} \\
&= e^{-\lambda}\Big[\sum_{k=2}^{\infty}\frac{k(k-1)\lambda^k}{k!} + \sum_{k=0}^{\infty}\frac{k\lambda^k}{k!}\Big] \\
&= e^{-\lambda}\Big[\lambda^2\sum_{k=2}^{\infty}\frac{k(k-1)\lambda^{k-2}}{k!} + \lambda\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}\Big] \\
&= e^{-\lambda}\Big[\lambda^2\sum_{k=0}^{\infty}\frac{\lambda^k}{k!} + \lambda\sum_{k=0}^{\infty}\frac{\lambda^k}{k!}\Big] \\
&= \lambda^2\sum_{k=0}^{\infty} e^{-\lambda}\frac{\lambda^k}{k!} + \lambda\sum_{k=0}^{\infty} e^{-\lambda}\frac{\lambda^k}{k!} \\
&= \lambda^2 + \lambda
\end{aligned}
$$

Now we can solve for variance by plugging that value into our equation.

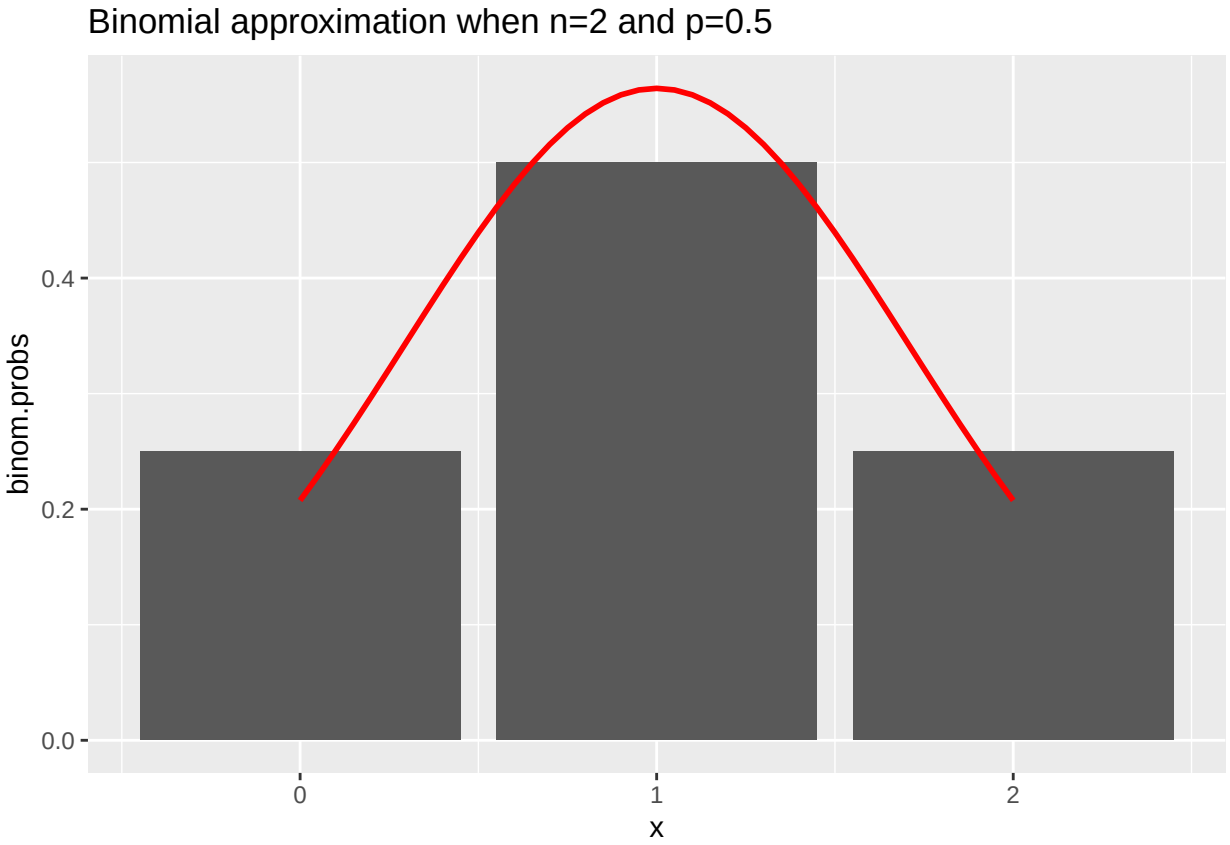
$$\boxed{Var(X) = E[X^2] - E[X]^2 = (\lambda^2 + \lambda) - (\lambda)^2 = \lambda}$$

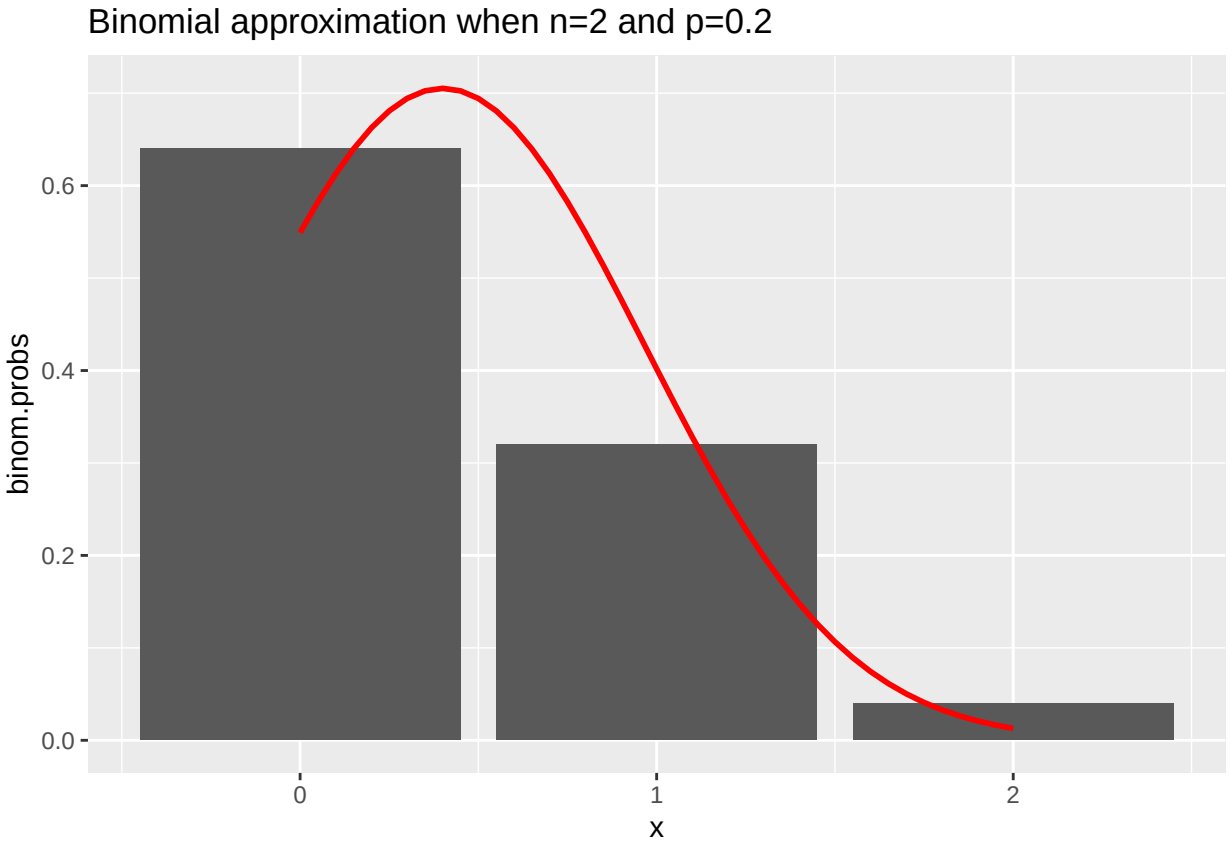
Question 2.b (5 points)

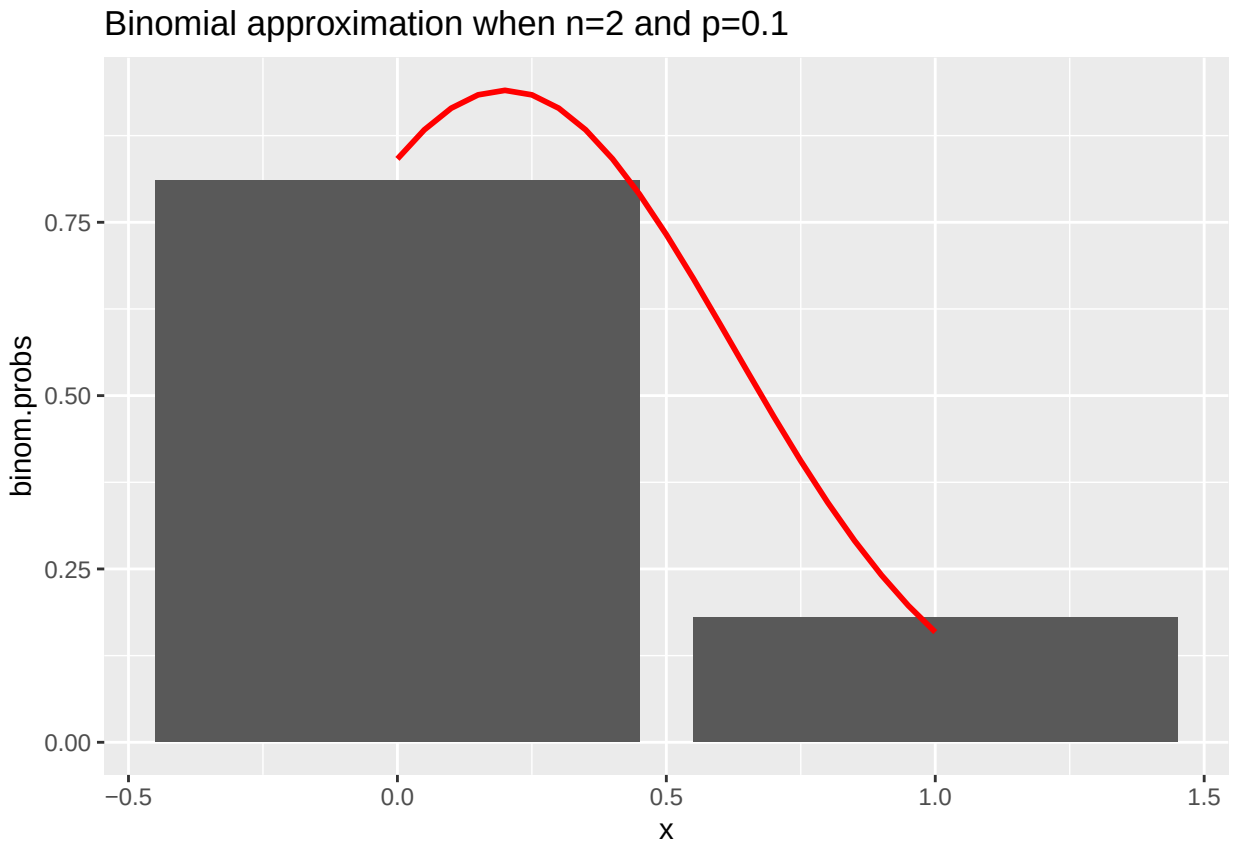
In this question, you will compare binomial distributions with Normal distributions having related means and variances.

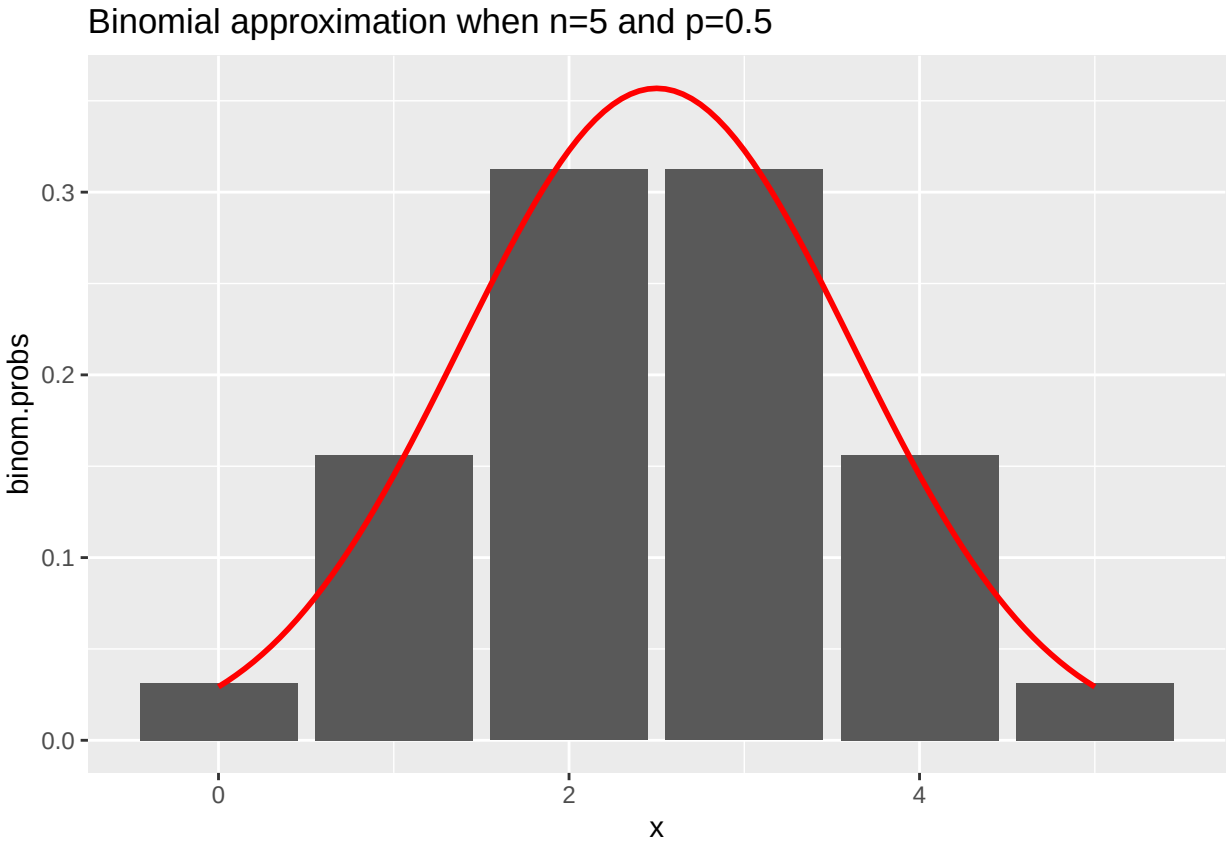
For each pair (n, p) with $n \in \{2, 5, 20\}$ and $p \in \{0.5, 0.2, 0.1\}$, please create a column plot over the non-negative integers less than or equal to n for which the quantile function of the $binomial(n, p)$ distribution is in $[.01, .99]$. Set the height of the column over the value k equal to the probability of k under the distribution $binomial(n, p)$. On this plot, draw the density of the Normal distribution with the same mean and variance as $binomial(n, p)$. Please label the plots with the corresponding n and p

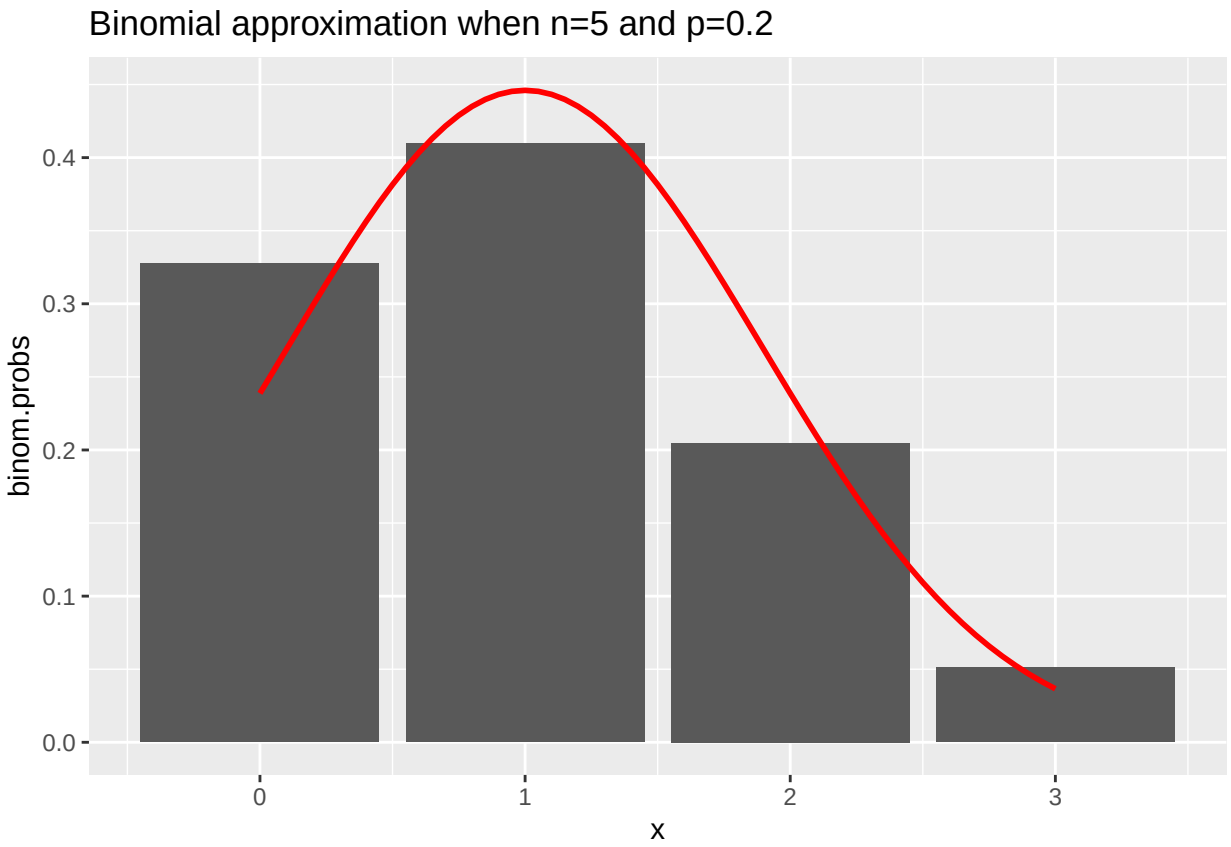
```
ns = c(2, 5, 20)
ps = c(0.5, 0.2, 0.1)

for (n in ns){
  for (p in ps){
    # Calculate the x-values at quanitles 0.01 and 0.99 to find the range of x values
    x.range = qbinom(c(0.01, 0.99), prob=p, size=n)
    # Get all values between the x range
    xs = x.range[1]:x.range[2]
    xs.cont = seq(x.range[1], x.range[2], 0.05)

    # Calculate the probability of that many successes for each x
    binom.probs = dbinom(xs, size=n, prob=p)
    # Calculate the normal approximation
    norm.probs = dnorm(xs.cont, mean=(n*p), sd=sqrt(n*p*(1-p)))

    # Plot the results in a column plot
    df.binom = data.frame(x=xs, prob=binom.probs)
    df.norm = data.frame(x=xs.cont, prob=norm.probs)
    title = sprintf("Binomial approximation when n=%s and p=%s", n, p)
    g <- ggplot(data=df.binom, aes(x=x, y=binom.probs)) +
          geom_bar(stat="identity") +
          geom_line(data=df.norm, aes(x=x, y=prob), color="red", size=1) +
          labs(title=title)
    print(g)
  }
}
```

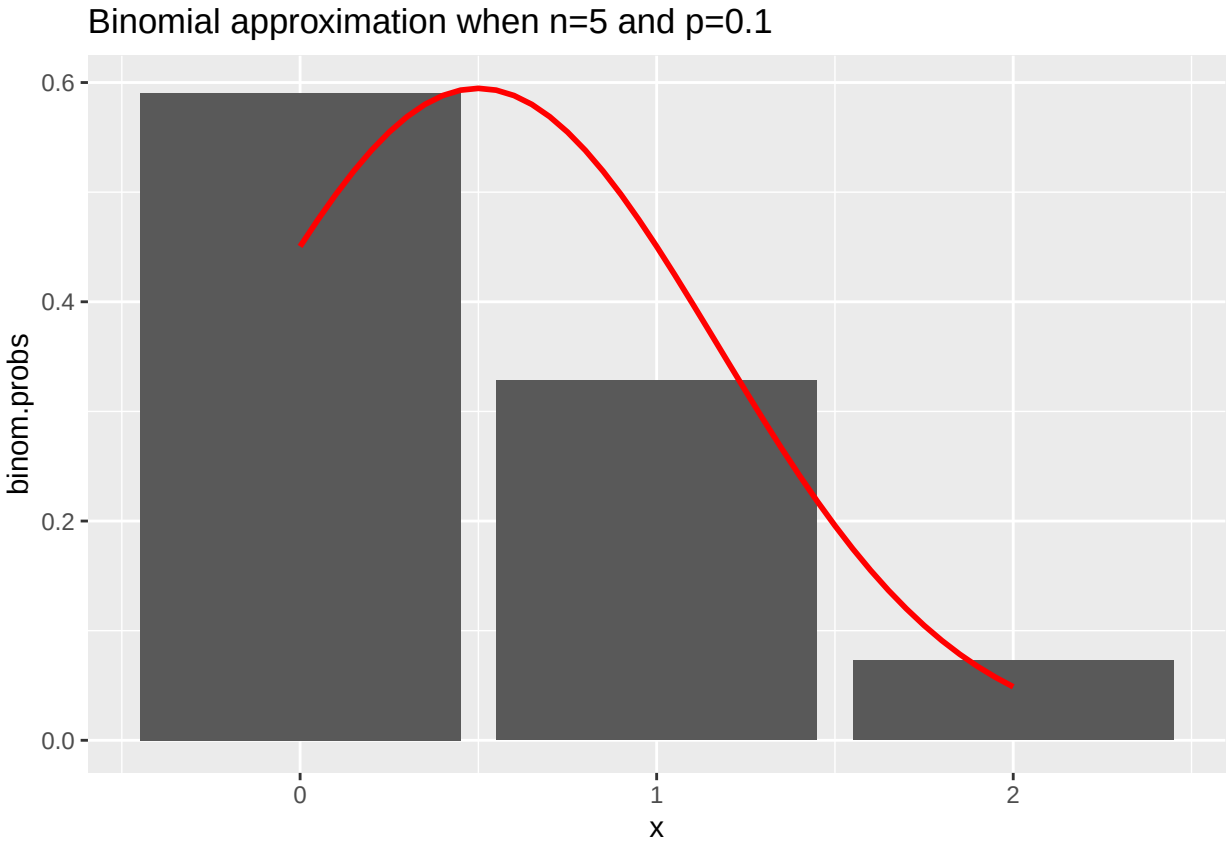

Binomial approximation when n=2 and p=0.5
binom.probs
0.4
0.2
0.0
0
1
2
x

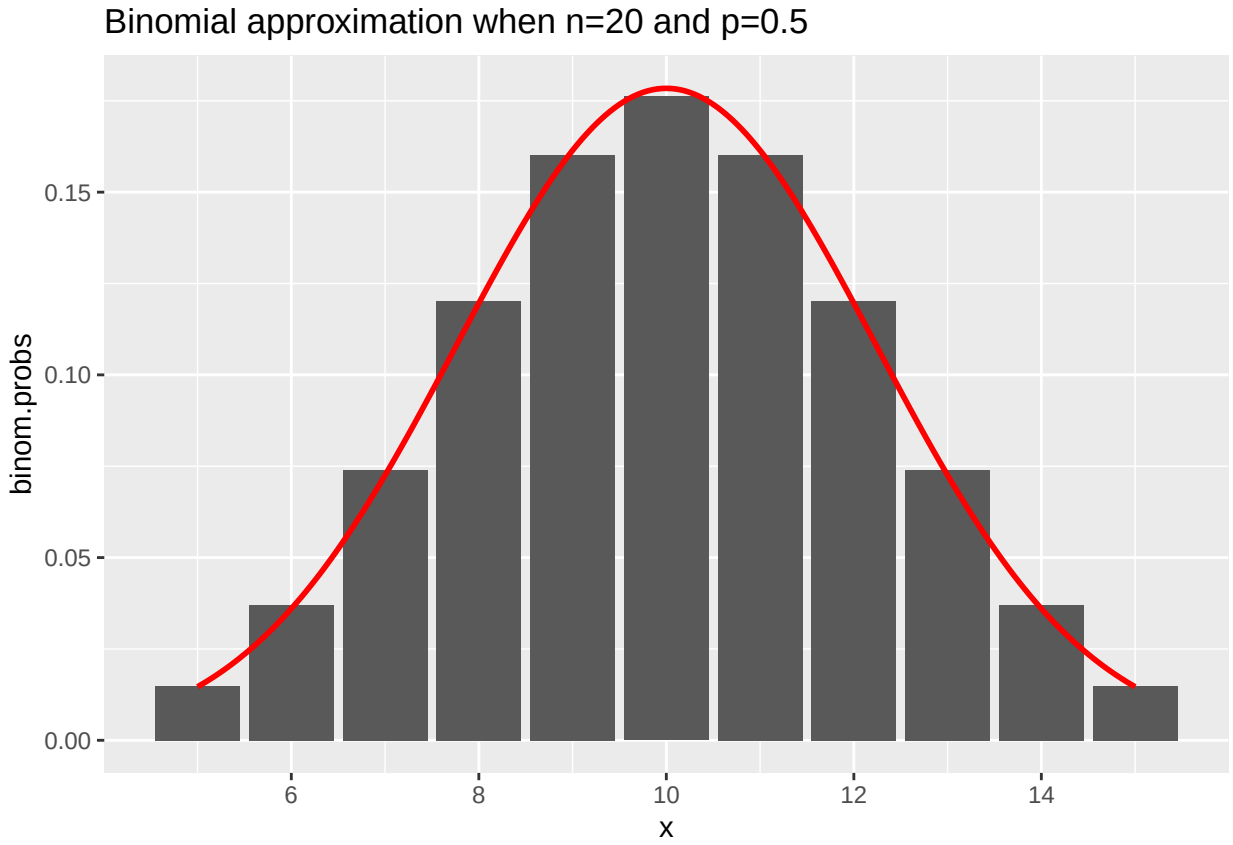

Binomial approximation when n=2 and p=0.2
binom.probs
0.0
0.2
0.4
0.6
0
1
2
x

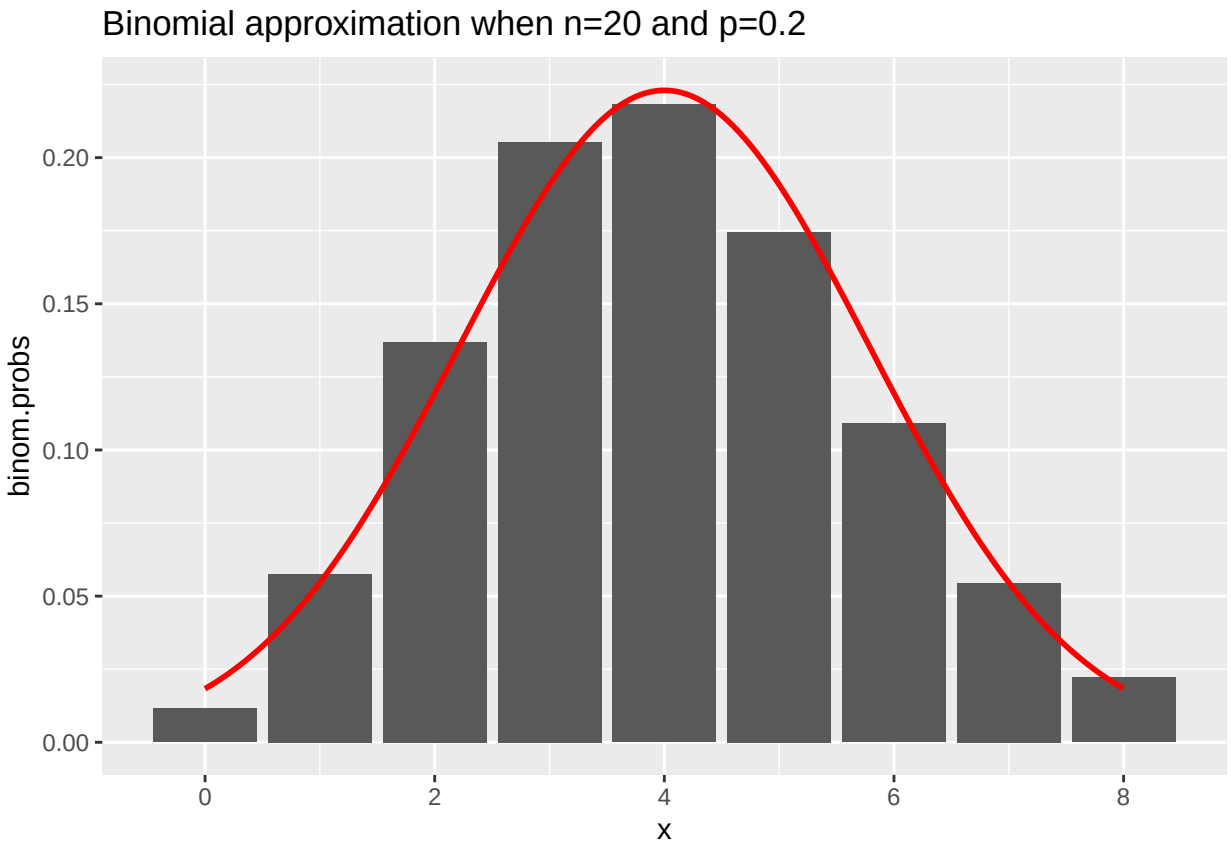

Binomial approximation when n=2 and p=0.1
binom.probs
0.75
0.50
0.25
0.00
−0.5
0.0
0.5
1.0
1.5
x

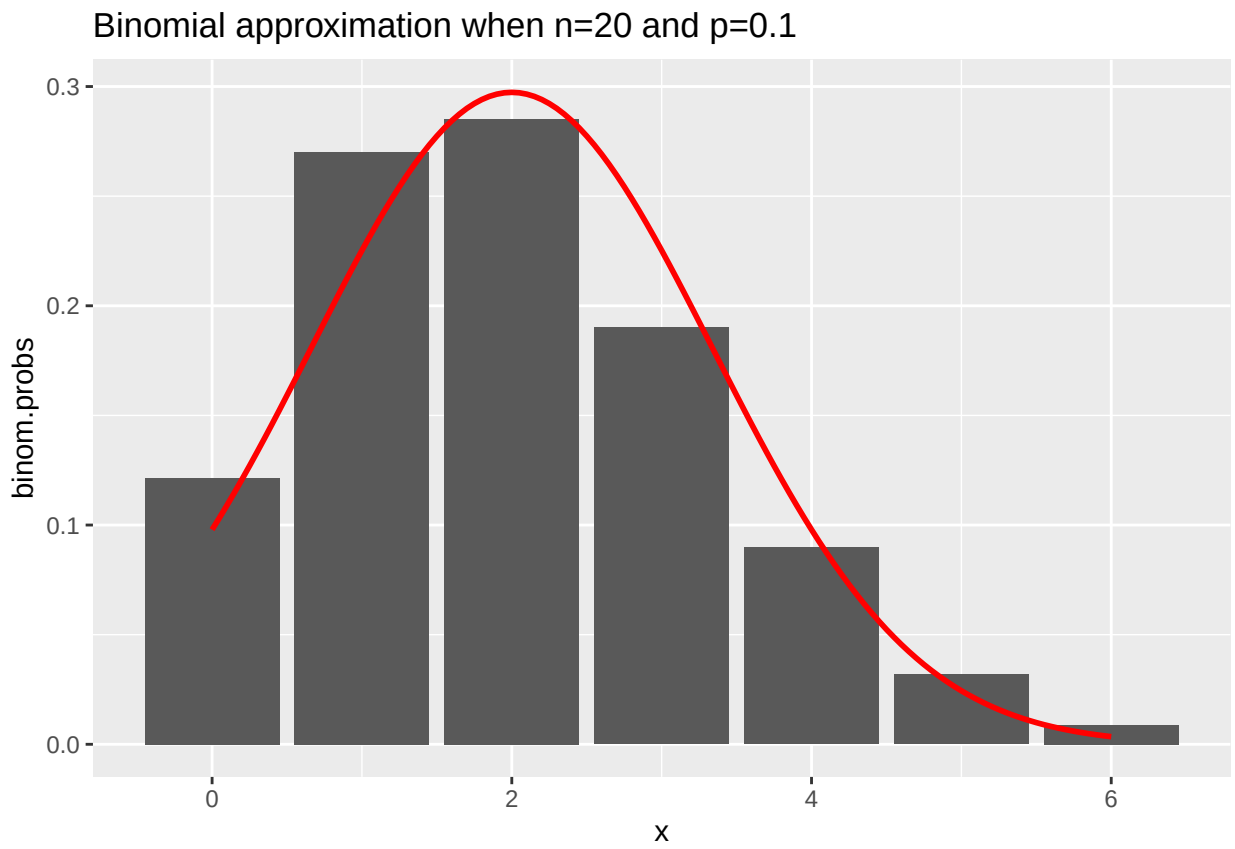

Binomial approximation when n=5 and p=0.5
binom.probs
0.0
0.1
0.2
0.3
0
2
4
x


Binomial approximation when n=5 and p=0.2
binom.probs
0.0
0.1
0.2
0.3
0.4
0
1
2
3
x


Binomial approximation when n=5 and p=0.1
0.6
0.4
0.2
0.0
binom.probs
0
1
2
x


Binomial approximation when n=20 and p=0.5
binom.probs
0.15
0.10
0.05
0.00
6
8
10
12
14
x


Binomial approximation when n=20 and p=0.2
binom.probs
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
x



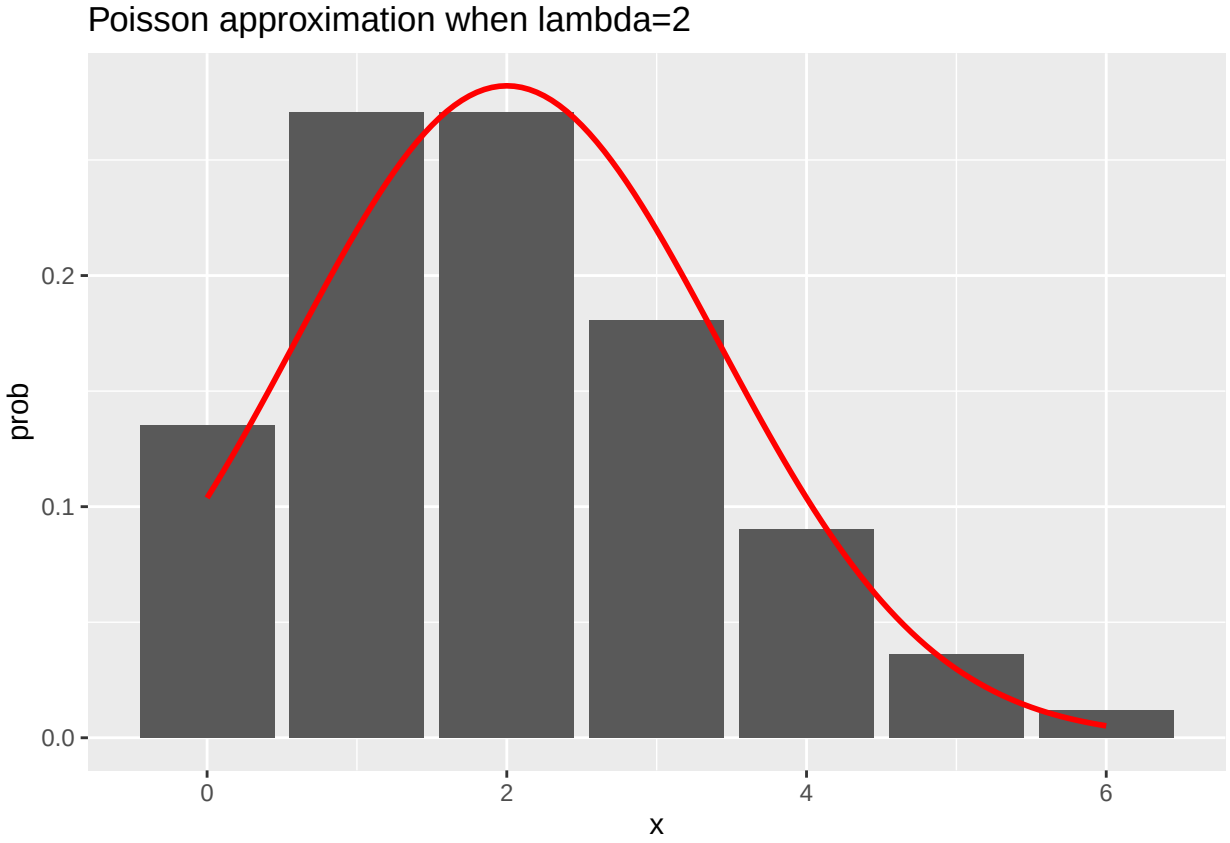
Question 2.c (5 points)

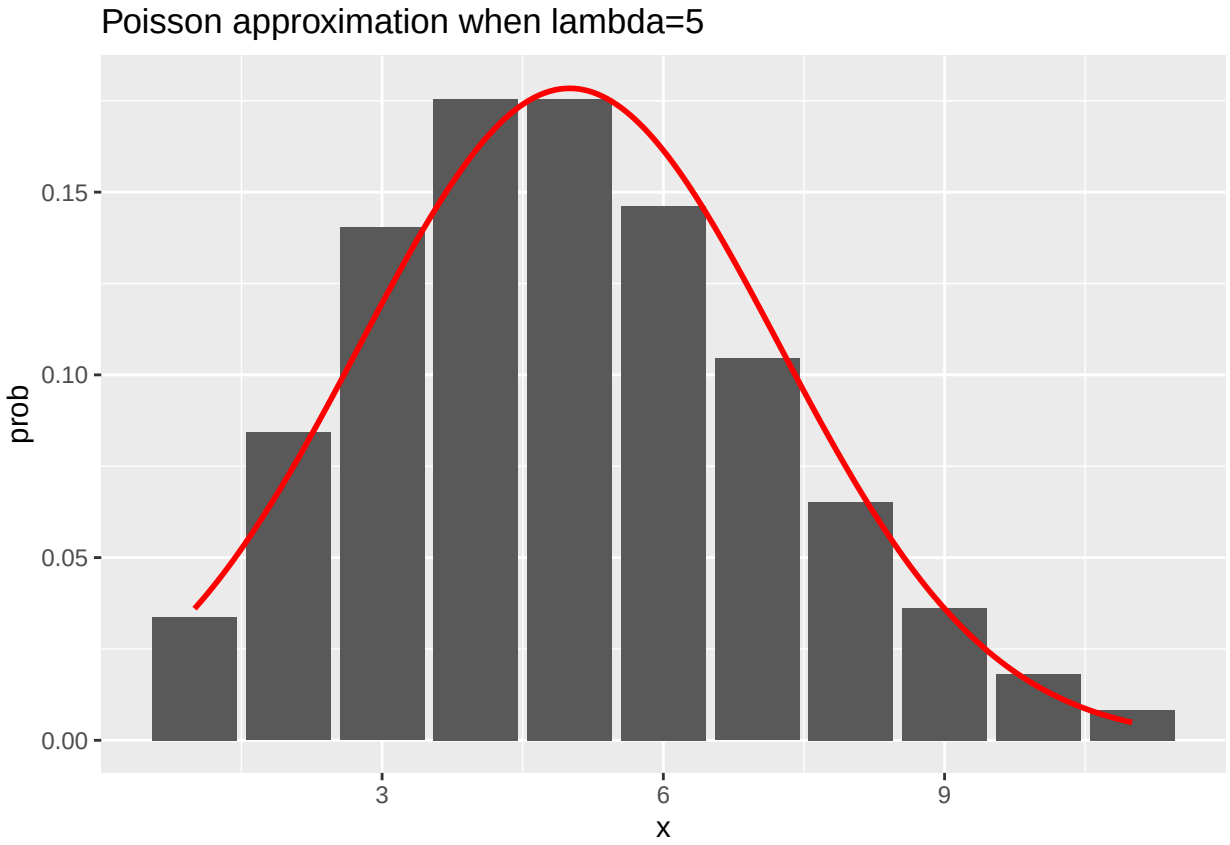
In this question, you will compare Poisson distributions with Normal distributions having related means and variances.

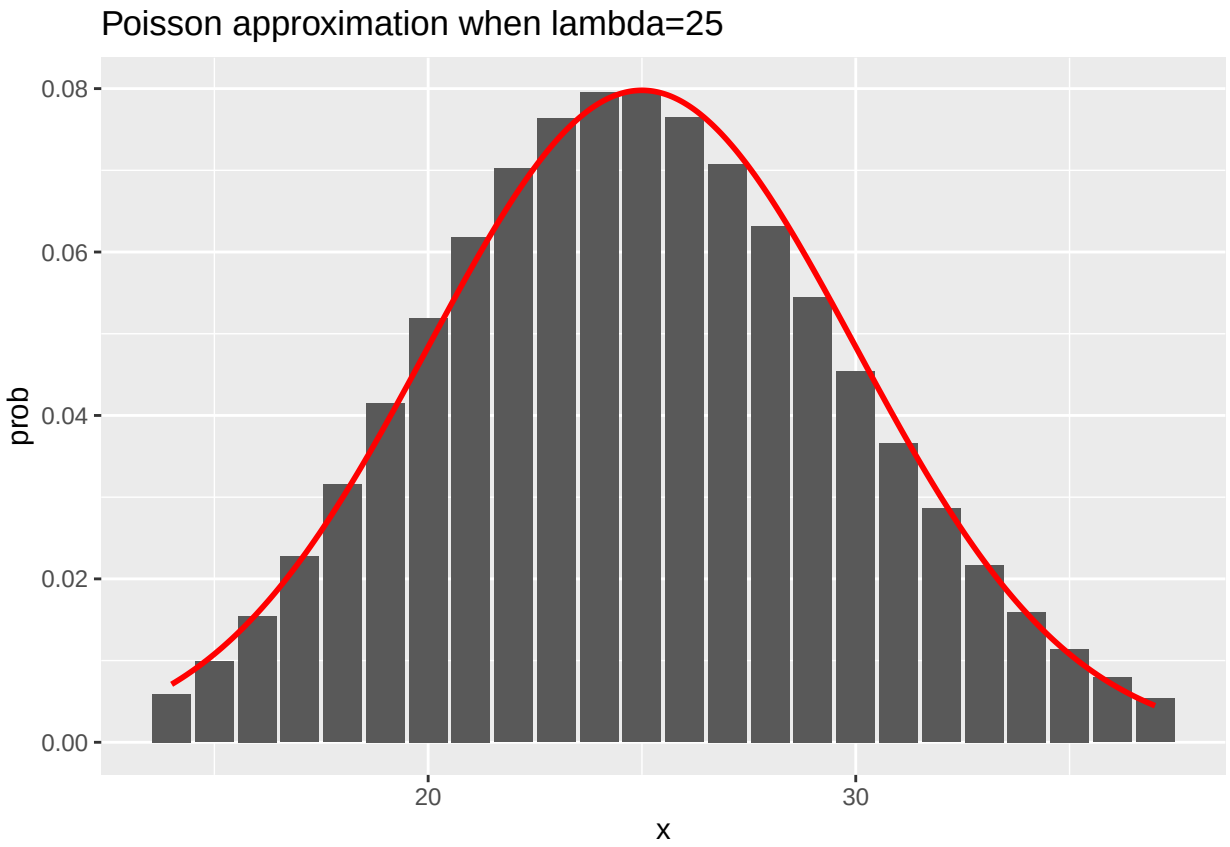
For values $\lambda \in \{2, 5, 25\}$, please create a column plot over the non-negative integers k for which the quantile function of the $Poisson(\lambda)$ distribution is in $[.01, .99]$. Set the height of the column over the value k equal to the probability of k under the distribution $Poisson(\lambda)$. On this plot, draw the density of the Normal distribution with the mean and variance both equal to λ. Please label the plots with the corresponding value of λ.

```
lambdas = c(2, 5, 25)

for (l in lambdas){
    # Calculate the x-values at quanitles 0.01 and 0.99 to find the range of x values
    x.range = qpois(c(0.01, 0.99), lambda=l)
    # Get all values between the x range
    xs = x.range[1]:x.range[2]
    xs.cont = seq(x.range[1], x.range[2], 0.05)

    # Calculate the probability of that many successes for each x
    pois.probs = dpois(xs, lambda=l)
    # Calculate the normal approximation
    norm.probs = dnorm(xs.cont, mean=l, sd=sqrt(l))
```

```
    # Plot the results in a column plot
    df.pois = data.frame(x=xs, prob=pois.probs)
    df.norm = data.frame(x=xs.cont, prob=norm.probs)
    title = sprintf("Poisson approximation when lambda=%s", l)
    g <- ggplot(data=df.pois, aes(x=x, y=prob)) +
          geom_bar(stat="identity") +
          geom_line(data=df.norm, aes(x=x, y=prob), color="red", size=1) +
          labs(title=title)
    print(g)
}
```




Poisson approximation when lambda=5
prob
0.15
0.10
0.05
0.00
3
6
9
x



Question 3

This problem simulates the situation in which a large number of researchers each draw samples of a specified size from a population and count the number of successes in the sample. Each researcher tests the hypothesis that the number of successes in their sample is consistent with the null model that the number of successes has a binomial distribution with size equal to the sample size and the probability of success equal to a specified value *pr*.

You will simulate the results in the case in which the samples are in fact drawn from the null distribution, the binomial distribution with size equal to the sample size and the probability of success equal to *pr*.

Question 3.a (5 points)

Assume each researcher uses the null model that their sample is sample from a binomial distribution with the size parameter equal to the number of items in the sample and the parameter giving the probability success equal to the value *pr*. One way to define a p-value for this null hypothesis is to calculate the twice probability of a value less than or equal to the observed number of successes under the null hypothesis, to calculate twice the probability of a value greater than or equal to the observed number of successes under the null hypothesis, and to take the smaller of these to be the p-value.

Please complete the function template below to make a p-value calculator for the researchers. The function should return the p-value of this test of the null hypothesis if the argument "obs" is the observed number of successes, the parameter "size" is the sample size, and the parameter "pr" is the probability of success under the null hypothesis. For future use, you may wish to implement this so that, given a vector of observations, the function returns a vector of the corresponding p-values.

Please apply your function to the case of an observation of 25 successes in a sample of size 100 with hypothesized probability of success equal to .3 and to an observation of 30 successes in a sample of size 100 with hypothesized probability of success equal to .3

```
# p-value calculator

p.get<-function(obs, size, pr){
  # Calculate the probability of observing that many or fewer successes
  pval.left = pbinom(obs, size=size, prob=pr)
  # Calculate the probability of observing that many or more successes
  # P(x>=k) = 1 - P(x<k) = 1-P(x<=k-1) = 1-pbinom(k-1)
  pval.right = 1 - pbinom(obs-1, size=size, prob=pr)
  # Note that pval.left + pval.right != 1 because both include the observed value

  # Get the minimum of each and return that value
  m <- matrix(c(pval.left, pval.right), nrow=2, byrow=TRUE)
  pval <- apply(m, 2, min)
  # Want twice the probability
  return (2*pval)
}

p.get(c(25,30),100,.3)
```

```
## [1] 0.3262602 1.0753205
```

Question 3.b (5 points)

Write a function with the arguments "n" for the number of researchers, "size" for the sample size, "pr" for the probability of success, and "p" for the p-value. The function should draw "n" samples from the binomial distribution with size equal "size" and probability of success equal to "pr". It should return the number of p-values less that or equal to "p" for the test above of the null hypothesis that the sample comes from a binomial distribution with size equal "size" and probability of success equal to "pr". (These are called *type 1 errors* at the significance level "p".) Apply "replicate" to this function 1000 times and calculate the mean number of type 1 errors for the sets of values

n=200,size=100,pr=.3,p=.05

n=200,size=500,pr=.3,p=.05

n=400,size=500,pr=.3,p=.01

n=400,size=1000,pr=.3,p=.01

What do the numbers computed represent in terms of the conclusion drawn by each of the 200 or 400 research teams?

```
num.t1.error <- function(n, size, pr, p){
    # Draw n samples from the binomial distribution with appropriate params
    binom.samples = rbinom(n, size=size, prob=pr)
    # Calculate the p-value for each number of observations
    pvals = p.get(binom.samples, size, pr)
    # Calculate the number of pvals that are <= p.
    num.t1error = sum(pvals <= p)
    return(num.t1error)
}
```

```r
mean(replicate(1000, num.t1.error(200, 100, 0.3, 0.05)))
```

```
## [1] 7.377
```

```r
mean(replicate(1000, num.t1.error(200, 500, 0.3, 0.05)))
```

```
## [1] 8.816
```

```r
mean(replicate(1000, num.t1.error(400, 500, 0.3, 0.01)))
```

```
## [1] 3.4
```

```r
mean(replicate(1000, num.t1.error(400, 1000, 0.3, 0.01)))
```

```
## [1] 3.426
```

The function returns the number of research teams that would reject the null hypothesis. By taking the mean of many simulations, we can get a better idea of how many will incorrectly reject the null when they shouldn't have. We can see that with a smaller p-value threshold, less teams encounter the Type 1 error, even as there were more teams in the overall sample.