Jiancong_Zhu

Instructions

R markdown is a plain-text file format for integrating text and R code, and creating transparent, reproducible and interactive reports. An R markdown file (.Rmd) contains metadata, markdown and R code "chunks", and can be "knit" into numerous output types. Answer the test questions by adding R code to the fenced code areas below each item. Once completed, you will "knit" and submit the resulting .html file, as well the .Rmd file. The .html will include your R code *and* the output.

Before proceeding, look to the top of the .Rmd for the (YAML) metadata block, where the *title* and *output* are given. Please change *title* from 'Programming with R Test #2' to your name, with the format 'lastName_firstName.'

If you encounter issues knitting the .html, please send an email via Canvas to your TA.

Each code chunk is delineated by six (6) backticks; three (3) at the start and three (3) at the end. After the opening ticks, arguments are passed to the code chunk and in curly brackets. **Please do not add or remove backticks, or modify the arguments or values inside the curly brackets**.

Depending on the problem, grading will be based on: 1) the correct result, 2) coding efficiency and 3) graphical presentation features (labeling, colors, size, legibility, etc). I will be looking for well-rendered displays. In the "knit" document, only those results specified in the problem statements should be displayed. For example, do not output - i.e. send to the Console - the contents of vectors or data frames unless requested by the problem. You should be able to display each solution in fewer than ten lines of code.

Submit both the .Rmd and .html files for grading.

Please delete the Instructions shown above prior to submitting your .Rmd and .html files.

Questions start here

Section 1: (15 points)

(1) R has probability functions available for use (Kabacoff, Section 5.2.3). Using one distribution to approximate another is not uncommon. (1)(a) (6 points) The Poisson distribution may be used to approximate the binomial distribution if $n > 20$ and $np < 7$. Estimate the following binomial probabilities using *dpois()* or *ppois()* with probability $p = 0.05$, and $n = 100$. Then, estimate the same probabilities using *dbinom()* or *pbinom()*. Show the numerical results of your calculations.

(i) The probability of exactly 0 successes.

```
n <- 100
p <- 0.05
lambda <- n*p

dpois(0, lambda)
```

```
## [1] 0.006737947
```

```
dbinom(0, n, p)
```

```
## [1] 0.005920529
```

(ii) The probability of fewer than 7 successes. Please note the following, taken from the Binomial Distribution R Documentation page, regarding the "lower.tail" argument:

lower.tail logical; if TRUE (default), probabilities are P[X ??? x], otherwise, P[X > x].

```
# Using Poisson approximation
ppois(6, lambda, lower.tail = TRUE)
```

```
## [1] 0.7621835
```

```
# Using Binomial distribution
pbinom(6, n, p, lower.tail = TRUE)
```

```
## [1] 0.766014
```

The binomial may also be approximated via the normal distribution. Estimate the following binomial probabilities using *dnorm()* or *pnorm()*, this time with probability p = 0.2 and n = 100. Then, calculate the same probabilities using *dbinom()* and *pbinom()*. Use continuity correction. Show the numerical results of your calculations.

(iii) The probability of exactly 25 successes.

```
p <- 0.2
n <- 100
mean <- n * p
sd <- sqrt(n * p * (1 - p))


# Using Normal approximation
# Applying continuity correction: P(X = 25) is approximated by P(24.5 < X < 25.5)
pnorm(25.5, mean, sd) - pnorm(24.5, mean, sd)
```

```
## [1] 0.04572879
```

```
# Using Binomial distribution
dbinom(25, n, p)
```

```
## [1] 0.04387783
```

(iv) The probability of fewer than 25 successes. Please note the following, taken from the Normal Distribution R Documentation page, regarding the "lower.tail" argument:

lower.tail logical; if TRUE (default), probabilities are P[X ??? x], otherwise, P[X > x].

```r
# Using Normal approximation with continuity correction: P(X < 25) is approximated by P(X < 24.5)
pnorm(24.5, mean, sd, lower.tail = TRUE)
```

```
## [1] 0.8697055
```

```r
# Using Binomial distribution
pbinom(24, n, p, lower.tail = TRUE)
```

```
## [1] 0.8686468
```

(1)(b) (3 points) Generate side-by-side barplots using *par(mfrow = c(1,2))* or *grid.arrange()*. The left barplot will show Poisson probabilties for outcomes ranging from 0 to 10. The right barplot will show binomial probabilities for outcomes ranging from 0 to 10. Use p = 0.1 and n = 100. Title each plot, present in color and assign names to the bar; i.e. x-axis value labels.

```r
# Define parameters for Poisson and Binomial distributions
lambda <- 10
n <- 100
p <- 0.1

# Generate probabilities for outcomes 0 to 10
poisson_probs <- dpois(0:10, lambda = lambda)
poisson_probs
```

```
##  [1] 4.539993e-05 4.539993e-04 2.269996e-03 7.566655e-03 1.891664e-02
##  [6] 3.783327e-02 6.305546e-02 9.007923e-02 1.125990e-01 1.251100e-01
## [11] 1.251100e-01
```

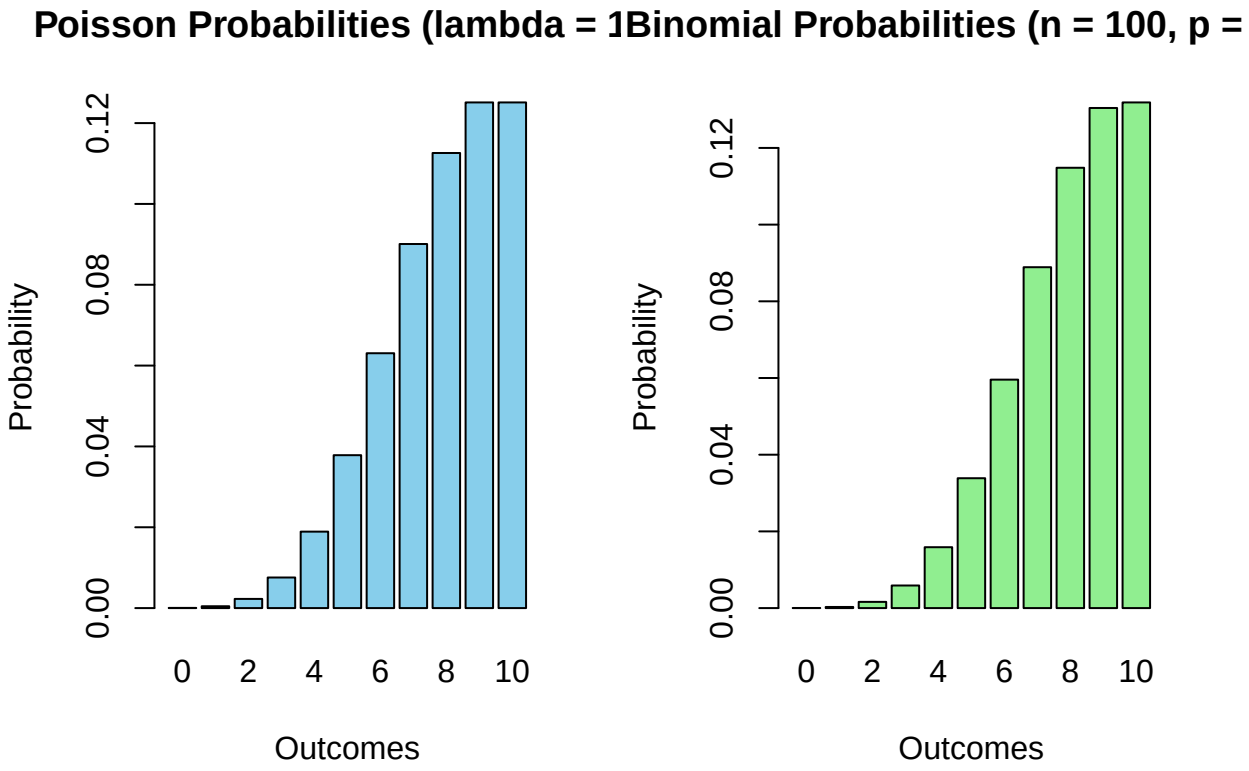
```r
binom_probs <- dbinom(0:10, size = n, prob = p)
binom_probs
```

```
##  [1] 0.0000265614 0.0002951267 0.0016231966 0.0058916025 0.0158745955
##  [6] 0.0338658038 0.0595787289 0.0888952464 0.1148230266 0.1304162771
## [11] 0.1318653468
```

```r
# Set up side-by-side barplots
par(mfrow = c(1, 2))

# Poisson barplot
barplot(poisson_probs, col = "skyblue", main = "Poisson Probabilities (lambda = 10)",
        names.arg = 0:10, xlab = "Outcomes", ylab = "Probability")

# Binomial barplot
barplot(binom_probs, col = "lightgreen", main = "Binomial Probabilities (n = 100, p = 0.1)",
        names.arg = 0:10, xlab = "Outcomes", ylab = "Probability")
```



(1)(c) (6 points) For this problem, refer to Sections 5.2 of Business Statistics. A discrete random variable has outcomes: 0, 1, 2, 3, 4, 5, 6. The corresponding probabilities in sequence with the outcomes are: 0.215, 0.230, 0.240, 0.182, 0.130, 0.003, 0.001. In other words, the probabilty of obtaining "0" is 0.215.

(i) Calculate the expected value and variance for this distribution using the general formula for mean and variance of a discrete distribution. To do this, you will need to use integer values from 0 to 6 as outcomes along with the corresponding probabilities. Round your answer to 1 decimal place.

```
# Define outcomes and probabilities
outcomes <- 0:6
probabilities <- c(0.215, 0.230, 0.240, 0.182, 0.130, 0.003, 0.001)

# Calculate expected value
expected_value <- sum(outcomes * probabilities)

# Calculate variance
variance <- sum((outcomes - expected_value)^2 * probabilities)

# Round results to 1 decimal place
round(expected_value, 1)
```

```
## [1] 1.8
```

```
round(variance, 1)
```

```
## [1] 1.8
```

(ii) Use the *cumsum()* function and plot the cumulative probabilties versus the corresponding outcomes. Determine the value of the median for this distribution and show on this plot. Note that there are methods for interpolating a median. However, we can identify an appropriate median from our set of our outcomes - 0 through 6 - that satisfies the definition. Creating a stair-step plot of the cumulative probability as a function of the outcomes may be helpful in identifying it.

```
# Define outcomes and probabilities
outcomes <- 0:6
probabilities <- c(0.215, 0.230, 0.240, 0.182, 0.130, 0.003, 0.001)

# Calculate expected value
expected_value <- sum(outcomes * probabilities)

# Calculate variance
variance <- sum((outcomes - expected_value)^2 * probabilities)

# Round results to 1 decimal place
expected_value <- round(expected_value, 1)
variance <- round(variance, 1)

expected_value
```

```
## [1] 1.8
```

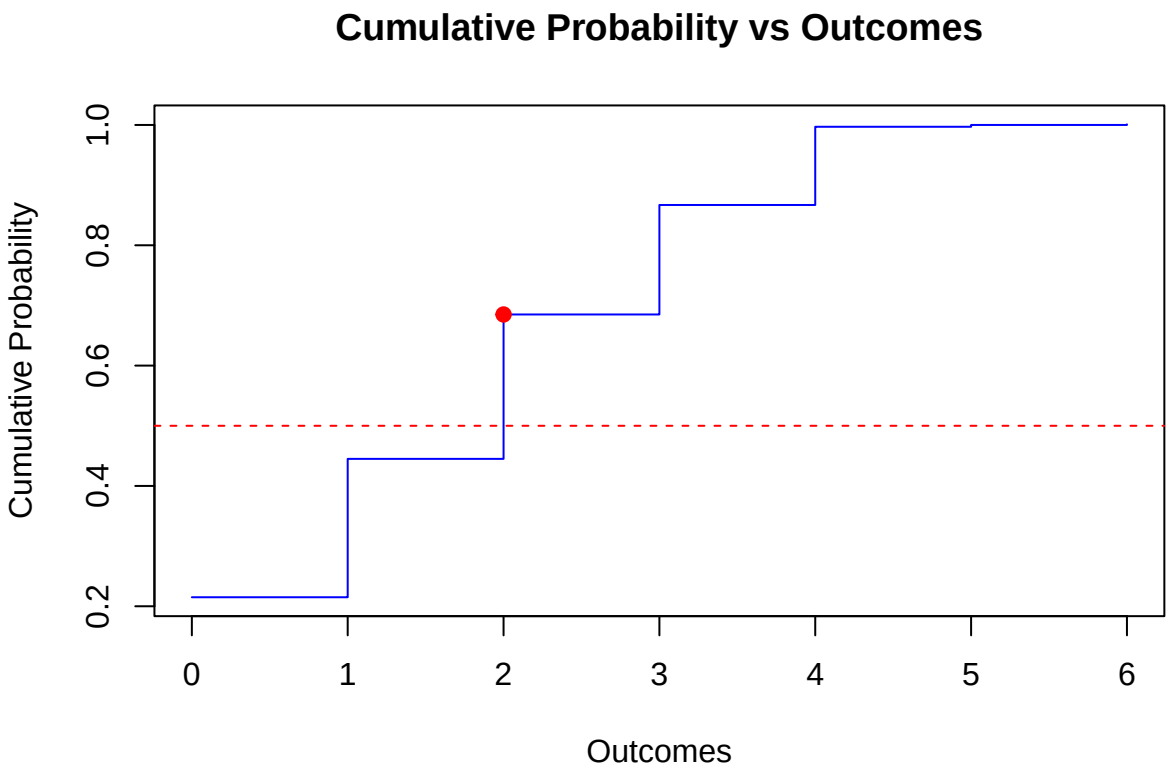
```
variance
```

```
## [1] 1.8
```

```
# Calculate cumulative probabilities
cumulative_probs <- cumsum(probabilities)

# Plot cumulative probabilities versus outcomes
plot(outcomes, cumulative_probs, type = "s", col = "blue", main = "Cumulative Probability vs Outcomes",
     xlab = "Outcomes", ylab = "Cumulative Probability")
abline(h = 0.5, col = "red", lty = 2)

# Determine median
median_value <- outcomes[which(cumulative_probs >= 0.5)[1]]
points(median_value, cumulative_probs[which(cumulative_probs >= 0.5)[1]], col = "red", pch = 19)
```




```
median_value
```

```
## [1] 2
```

Section 2: (15 points)

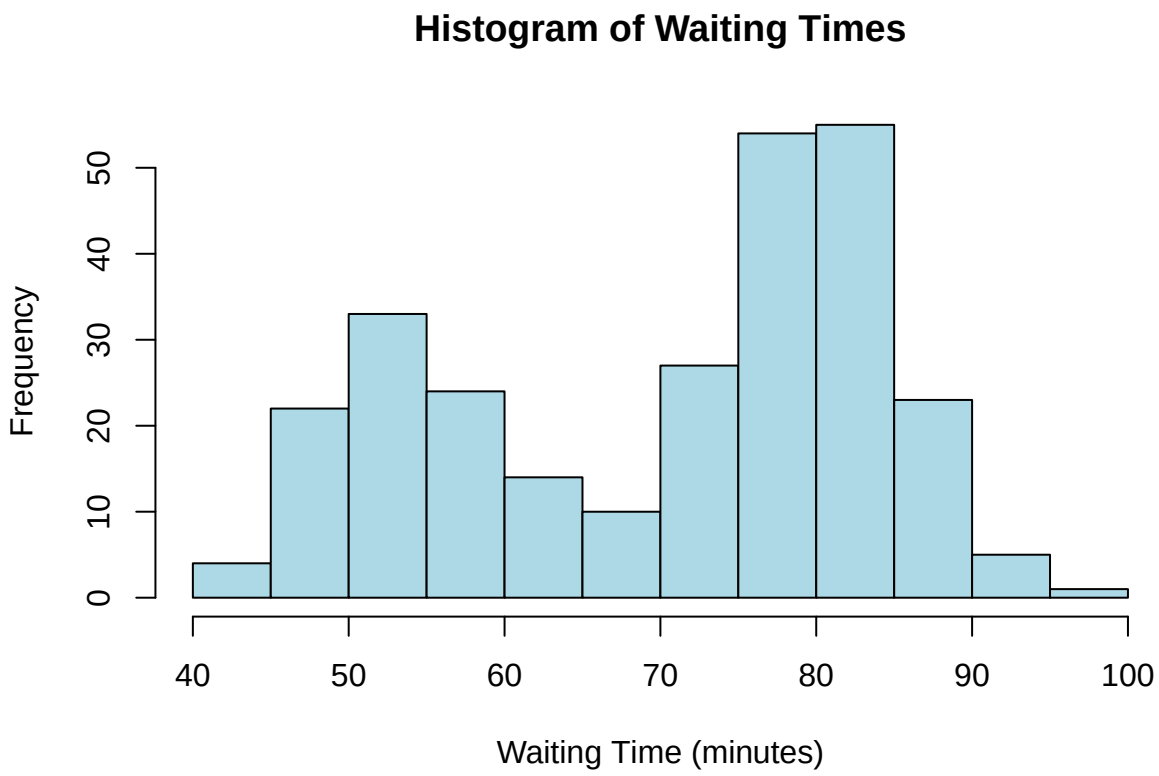
(2) Conditional probabilities appear in many contexts and, in particular, are used by Bayes' Theorem. Correlations are another means for evaluating dependency between variables. The dataset "faithful" " is part of the "datasets" package and may be loaded with the statement *data(faithful)*. It contains 272 observations of 2 variables; waiting time between eruptions (in minutes) and the duration of the eruption (in minutes) for the Old Faithful geyser in Yellowstone National Park. (2)(a) (6 points) Load the "faithful" dataset and present summary statistics and a histogram of waiting times. Additionally, compute the empirical conditional probability of an eruption less than 3.5 minutes, if the waiting time exceeds 70 minutes.

```
data(faithful, package = "datasets")

# Summary statistics for "waiting" time
summary_stats <- summary(faithful$waiting)
summary_stats
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    43.0    58.0    76.0    70.9    82.0    96.0
```

```r
# Histogram of waiting times
hist(faithful$waiting, col = "lightblue", main = "Histogram of Waiting Times",
     xlab = "Waiting Time (minutes)", ylab = "Frequency")
```




```r
# Empirical conditional probability of eruption < 3.5 minutes given waiting time > 70 minutes
subset_data <- faithful[faithful$waiting > 70, ]
conditional_prob <- mean(subset_data$eruptions < 3.5)

conditional_prob
```

```
## [1] 0.02424242
```

(i) Identify any observations in "faithful" for which the waiting time exceeds 90 minutes and the eruptions last longer than 5 minutes. List and show any such observations in a distinct color on a scatterplot of all eruption (vertical axis) and waiting times (horizontal axis). Include a horizontal line at eruption = 5.0, and a vertical line at waiting time = 90. Add a title and appropriate text.

```r
# Identify observations where waiting time > 90 and eruptions > 5
subset_observations <- faithful[faithful$waiting > 90 & faithful$eruptions > 5, ]
subset_observations
```

```
##     eruptions waiting
## 149       5.1      96
```

```r
# Scatterplot of all eruption and waiting times
plot(faithful$waiting, faithful$eruptions, main = "Eruption Duration vs Waiting Time",
     xlab = "Waiting Time (minutes)", ylab = "Eruption Duration (minutes)", pch = 16, col = "grey")

# Highlight specific observations in a distinct color
points(subset_observations$waiting, subset_observations$eruptions, col = "red", pch = 19)

# Add horizontal and vertical lines
abline(h = 5, col = "blue", lty = 2)
abline(v = 90, col = "blue", lty = 2)

# Add appropriate title and labels
text(95, 4, labels = "Waiting Time > 90 & Eruptions > 5", col = "red", pos = 4)
```