PSTAT 10 (Practice) Final Exam Solutions

Name:

TA:

You have 80 minutes to complete 6 problems, some of which contain subproblems.

You are allowed one sheet, two sides of handwritten notes.

If you run out of space for your code, write your code on the backs of the pages and clearly indicate which problem it corresponds to.

Please make sure your handwritten code is legible. I cannot give credit to code that is unreadable.

Problem 1: Bus stop (6 pts)

1. A bus containing m passengers arrives at a bus stop where 10 people are waiting. Each of the m passengers independently has a probability 0.10 of leaving the bus. Each person waiting at the bus stop independently has a 0.25 probability of boarding the bus. (1 pt)

What is the distribution of the number of passengers who will exit the bus?

$$\text{Binom}(m, 0.10)$$

What is the distribution of the number of new passengers who will board the bus?

$$\text{Binom}(10, 0.25)$$

2. A bus arrives at n bus stops $A_1, A_2, \ldots, A_n$ in order. The bus is empty to begin with and 10 people are waiting at each bus stop. After the bus arrives at a stop, the passengers currently on the bus independently have a probability of 0.10 of leaving the bus. After this, the 10 people waiting will each board the bus independently with probability 0.25.

Write a function `passenger(n)` that simulates the bus and returns the number of passengers remaining on the bus after leaving the n-th bus stop. (3 pts)

```r
passenger <- function(n) {
  num_pass <- 0
  for(i in seq_len(n)) {
    num_pass <- num_pass - rbinom(1, num_pass, 0.1)
    num_pass <- num_pass + rbinom(1, 10, 0.25)
  }
  return(num_pass)
}
```

3. Write code to simulate, using 1,000 replications, the expected number of passengers on the bus after 3 bus stops. (2 pts)

```r
mean(replicate(1000, passenger(3)))
```

```
## [1] 6.782
```

Problem 2: Poisson Distribution (6 pts)

How many emails will I receive in an hour? How many earthquakes will occur in a year in some region of the world? A popular probability distribution that models such events is the *Poisson distribution.* We write $X \sim \text{Pois}(\lambda)$ if X is a Poisson random variable with parameter λ, where λ is the *rate* of occurrence of an event given in emails/hour, earthquakes/year, etc.

R has built-in functions for the Poisson distribution. They are:

- `dpois(x, lambda)`
- `ppois(q, lambda)`
- `rpois(n, lambda)`

Let $X \sim \text{Pois}(2)$ be the number of emails I receive in an hour.

1. State the support of X. Is X continuous or discrete? (2 pts)

X is discrete with support on the nonnegative integers $0, 1, 2, \ldots$.

2. Determine the probability I receive greater than 5 but less than 10 emails in an hour. (2 pts)

```
ppois(9, 2) - ppois(5, 2)
```

```
## [1] 0.01651711
```

3. Determine the probability I receive no emails in an hour. (2 pts)

```
dpois(0, 2)
```

```
## [1] 0.1353353
```

Problem 3: Majority (5 pts)

Write a function `majority(v)` that takes an numeric vector and returns the *majority element*; i.e., the element that appears more than `n/2` times, where `n` is the length of `v`. The input `v` is guaranteed to be nonempty and contain positive integers. A majority element of `v` is guaranteed to exist.

```r
majority <- function(v) {
  v[which(tabulate(v) > length(v) / 2)]
}

majority(c(3, 2, 3))
```

```
## [1] 3
```

```r
majority(c(2, 2, 1, 1, 1, 2, 2))
```

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

Hint: You may use the `tabulate(v)` function, which returns a vector of the counts of elements of a numeric vector `v` in the position indexed by that element.

```r
# Each number from 1 through 3 appears once, 5 appears once, but 4 appears zero times.
tabulate(c(1:3, 5))
```

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

```r
# 1 appears zero times, 2 appears two times, 3 appears one time,
# 4 appears zero times, 5 appears two times.
tabulate(c(2, 2, 3, 5, 5))
```

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

Problem 4: Second largest (5 pts)

Write a function `second_largest(vec)` that takes a numeric vector `vec` and returns its second largest element. The elements of `vec` are guaranteed to be nonnegative and unique.

```r
second_largest <- function(vec) {
  largest <- -1
  sec_largest <- -1
  for (v in vec) {
    if (v > largest) {
      sec_largest <- largest
      largest <- v
    }
  }
  sec_largest
}

second_largest(1:10)
```

```
## [1] 9
```

```
second_largest(c(5, 0, 6.6, 3.13, 1.5))
```

```
## [1] 5
```

Problem 5: Some data (6 pts)

The tibble `some_data` contains some randomly generated values.

```
some_data
```

```
## # A tibble: 100 x 3
##        a      b c
##    <int>  <dbl> <fct>
##  1     7 0.327  J
##  2     7 0.389  J
##  3     8 0.0411 I
##  4     5 0.361  K
##  5     8 0.571  J
##  6     8 0.685  I
##  7    10 0.971  J
##  8     7 0.702  K
##  9     8 0.0115 J
## 10     6 0.536  K
## # ... with 90 more rows
```

1. Create a tibble called `some_data1` of cases with `c` level of `K` and only variables `c` and `d` where `d` is the sum of variables `a` and `b`. Match the provided result exactly. (2 pts)

```
some_data1 <- some_data |>
                filter(c == "K") |>
                mutate(d = a + b) |>
                select(c, d)
some_data1
```

```
## # A tibble: 35 x 2
##    c         d
##    <fct> <dbl>
##  1 K      5.36
##  2 K      7.70
##  3 K      6.54
##  4 K      7.20
##  5 K      9.58
##  6 K      8.35
##  7 K     12.0
##  8 K      8.55
##  9 K      9.58
## 10 K      9.04
## # ... with 25 more rows
```

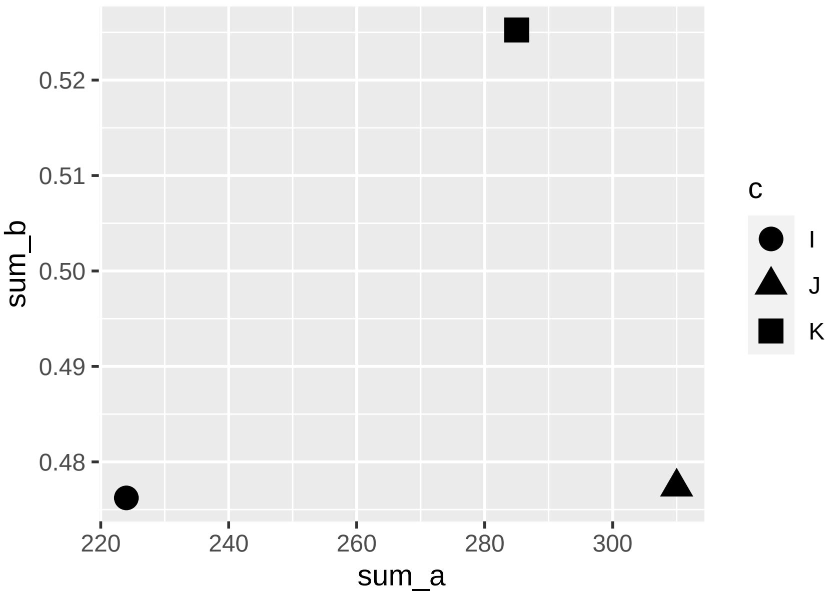
2. Create a tibble `some_data2` containing, for each level of `c`, the sum of `a` and the mean of `b` for that level of `c`. Match the provided result exactly. (2 pts)

```
(some_data2 <- some_data |>
                group_by(c) |>
                summarize(sum_a = sum(a), sum_b = mean(b)))
```

```
## # A tibble: 3 x 3
##   c     sum_a sum_b
##   <fct> <int> <dbl>
## 1 I       224 0.476
## 2 J       310 0.477
## 3 K       285 0.525
```

3. Using `some_data2`, create the ggplot that's given on the next page. (2 pts)

```
ggplot(some_data2, mapping = aes(x = sum_a, y = sum_b, shape = c)) +
  geom_point(size = 4) # size = 4 not required for credit. It's only there for clarity.
```



Problem 6: tinyclothes (8 pts)

The `tinyclothes` database stores information about customers buying products at a store. It contains four tables; `Customer` contains customer information, `SalesOrder` contains customer orders, `SalesOrderLine` is an association table between orders and products within an order, and `Product` contains product information.

The data are given below.

Customer

CustomerId	Name	Address
1	Alex	State
2	Bob	Hollister
3	Carol	Ocean
6	Juan	Phelps

SalesOrder

CustomerId	OrderId	Date
1	1	11/11/19
3	2	7/9/19
6	9	8/16/19
6	10	10/12/19

SalesOrderLine

OrderId	ProductId	Quantity
1	1	10
2	1	10
2	4	20
9	1	5
10	1	5

Product

ProductId	Name	Color
1	Pants	Blue
2	Pants	Khaki
3	Socks	Green
4	Socks	White
5	Shirts	White

Write SQL queries to answer the following questions. **For full credit, write a single query that matches the provided output.** Also, write only SQL and nothing else.

In these solutions, you should only write the SQL query and not R commands like `dbGetQuery`

For the purposes of studying, I've provided you the database online so you can test your query. There are multiple approaches. As long as your get the right answer it's right

1. Retrieve the CustomerId, Name, Address, OrderId, and Date for customer Alex. (2 pts)

```
select c.CustomerId, name, address, orderid, date from customer c
          inner join salesorder so on c.customerid = so.customerid
        where c.Name = 'Alex'
```

```
##   CustomerId Name Address OrderId     Date
## 1          1 Alex   State       1 11/11/19
```

2. Retrieve the CustomerId, Name, and total quantity of products ordered by customer Carol. (3 pts)

```
select c.CustomerId, c.Name, sum(Quantity) TotalQuantity from customer c
    inner join salesorder so on c.customerid = so.customerid
    inner join salesorderline sol on so.orderid = sol.orderid
  where name = 'Carol'
  group by c.CustomerId
```

```
##   CustomerId  Name TotalQuantity
## 1          3 Carol            30
```

3. Retrieve the *unique* CustomerId, customer name, ProductId, product name, and color for all customers who ordered Blue Pants. (3 pts)

```
select distinct c.customerid, c.name as CustomerName,
            p.productid, p.name as ProductName, p.color from customer c
          inner join salesorder so on c.customerid = so.customerid
          inner join salesorderline sol on so.orderid = sol.orderid
          inner join product p on sol.productid = p.productid
        where p.productid = 1
```

```
##   CustomerId CustomerName ProductId ProductName Color
## 1          1         Alex         1       Pants  Blue
## 2          3        Carol         1       Pants  Blue
## 3          6         Juan         1       Pants  Blue
```