ORIE 3120 Prelim 1

March 3, 2020

- The exam is 1.5 hours long.
- You may use one (8.5×11) sheet of notes (back and front).
- No computers, calculators, phones, or other electronic devices allowed.
- Use the space after each question for your answer. The backs of the exam pages may be used for scratchwork but will not be graded.

Name: _____
NetID: _____

Honor code. Write the following statement, and sign your name below:

I certify that all work on this exam is my own.

Honor code: _____
Signature: _____

1. *Spotify.*

Spotify Global Top 50 is ranked by the number of streams. We have the data for February 20, 2020 and February 19, 2020 in tables named Feb20 and Feb19, respectively. The following are the first 10 data entries from each table.

February 19:

	Position	Title	Artist	Streams
1	1	Blinding Lights	The Weeknd	5983876
2	2	Dance Monkey	Tones and I	5957301
3	3	The Box	Roddy Ricch	5749543
4	4	Intentions (feat. Quavo)	Justin Bieber	4833667
5	5	Don't Start Now	Dua Lipa	4633763
6	6	No Time To Die	Billie Eilish	4467091
7	7	Yummy	Justin Bieber	4037354
8	8	Tusa	KAROL G	4002459
9	9	Falling	Trevor Daniel	3871618
10	10	ROXANNE	Arizona Zervas	3714629

February 20:

	Position	Title	Artist	Streams
1	1	Blinding Lights	The Weeknd	6127151
2	2	Dance Monkey	Tones and I	5980798
3	3	The Box	Roddy Ricch	5643872
4	4	Intentions (feat. Quavo)	Justin Bieber	4753045
5	5	Don't Start Now	Dua Lipa	4601096
6	6	No Time To Die	Billie Eilish	4231654
7	7	After Hours	The Weeknd	4152572
8	8	Tusa	KAROL G	4067362
9	9	Yummy	Justin Bieber	3991664
10	10	Falling	Trevor Daniel	3821073

You may assume that each title appears only once in the chart (each track has a unique title), but there may be artists with multiple songs in the same chart.

(a) (5 points) You want to compare how the ranking changed from February 19 to February 20 for each song that was ranked on February 19. Create a query that has

- Column 1: February 19th ranking
- Column 2: February 20th ranking (or NULL if the song was not ranked on February 20)
- Column 3: Change in rankings

$$\begin{cases} 0 & \text{if the ranking did not change} \\ n & \text{if the ranking went up } n \text{ positions} \\ -n & \text{if the ranking went down n positions} \end{cases}$$

  (or NULL if the song was not ranked on February 20)
- Column 4: Title
- Column 5: Artist

The first 10 entries in the resulting view should look something like this:

	Feb19Position	Feb20Position	Difference	Title	Artist
1	1	1	0	Blinding Lights	The Weeknd
2	2	2	0	Dance Monkey	Tones and I
3	3	3	0	The Box	Roddy Ricch
4	4	4	0	Intentions (feat. Quavo)	Justin Bieber
5	5	5	0	Don't Start Now	Dua Lipa
6	6	6	0	No Time To Die	Billie Eilish
7	7	9	-2	Yummy	Justin Bieber
8	8	8	0	Tusa	KAROL G
9	9	10	-1	Falling	Trevor Daniel
10	10	11	-1	ROXANNE	Arizona Zervas

Solution:

```
SELECT Feb19.Position AS Feb19Position,
     Feb20.Position AS Feb20Position,
    (Feb19.Position - Feb20.Position) AS Difference,
     Title,
     Artist
FROM Feb19
LEFT JOIN Feb20 ON Feb19.Title = Feb20.Title
```

(b) (5 points) You want to see the songs for which the number of streams changed by less than 10,000. Create a query that has

- Column 1: Number of streams on February 19th
- Column 2: Number of streams on February 20th
- Column 3: Difference in number of streams on February 19th vs 20th
- Column 4: Title

- Column 5: Artist

ordered in descending order of the difference in number of streams. The resulting view should look something like this:

	Feb19streams	Feb20streams	Difference	Title	Artist
1	2603655	2596922	6733	bad guy	Billie Eilish
2	1822026	1817361	4665	Vete	Bad Bunny
3	2461242	2460221	1021	Adore You	Harry Styles
4	1568203	1567491	712	Fantasias	Rauw Alejandro
5	2058104	2057519	585	Physical	Dua Lipa
6	1890286	1891515	-1229	Lose Control	MEDUZA
7	2243397	2245634	-2237	RITMO (Bad Boys For Life)	Black Eyed Peas
8	2340900	2345883	-4983	Señorita	Shawn Mendes
9	2658748	2665035	-6287	Ignorantes	Bad Bunny
10	2100261	2110238	-9977	Blueberry Faygo	Lil Mosey

Hint: The absolute value of a number x is abs(x) in SQL

Solution:

```
SELECT Feb19.Streams AS Feb19streams,
     Feb20.Streams AS Feb20streams,
    (Feb19.Streams - Feb20.Streams) AS Difference,
     Title,
     Artist
FROM Feb19
LEFT JOIN Feb20 ON Feb19.Title = Feb20.Title
WHERE abs(Difference) <= 10000
ORDER BY Difference DESC
```

2. *Majors.* We have two databases, Students and Majors. The Students table has four columns Student ID, Name, Major and Year. You can assume each StudentID appears exactly once in the Students table (names may repeat) and that each student has exactly one major.

   The first three columns of the Students table are below.

StudentID	Name	Major	Year
1	James Smith	3	2
2	Maria Ramirez	2	2
3	Sarah Jones	7	3

   The Majors table has three columns, MajorID, MajorName, and NumberOfFaculty, which gives the number of faculty associated with the major. The first four columns of the Majors table are below.

MajorID	MajorName	NumberOfFaculty
1	American History	25
2	Computer Science	52
3	Math	23
4	Spanish	34

   (a) (5 points) You want to know which majors have the most students. Write a query that has:

   - Column 1: MajorID
   - Column 2: MajorName
   - Column 3: NumberofStudents

   The results should be in descending order of the number of students. Ensure all majors appear in the table.

   Solution:

```
SELECT MajorID, Major Name, Count(StudentID) AS NumberOfStudents
FROM Majors LEFT JOIN Students
ON Majors.MajorID = Students.Major
GROUP BY MajorName
ORDER BY NumberOfStudents DESC
```

   Note: Some majors may not be listed if you use INNER JOIN instead of LEFT JOIN

   (b) (5 points) You would also like to know the student to faculty ratio. Write a query, or explain how you would change to the previous query, to produce a column which gives the student to faculty ratio for each major. Call your column StudentFacultyRatio.

Solution: To the select statement add the following:

```
COUNT(StudentID)/NumberOfFaculty AS StudentFacultyRatio
```

(c) (5 points) Finally, you want a list of all the second-year students' names and their majors. Write a query that has:

- Column 1: StudentName
- Column 2: MajorName

Order your results alphabetically by StudentName

Solution:

```
SELECT Name, MajorName
FROM Students
LEFT JOIN
Majors ON Students.Major = Majors.MajorID
WHERE Year = 2
ORDER BY Name
```

Note: Since they are told each student has exactly one major, inner join and left join are equivalent.

3. *Piazza awards.* As a way to incentivize students to make good Piazza posts, the ORIE 3120 staff have decided to establish the "Piazza Awards" for the students with the most good posts on Piazza, with 1st place, 2nd place and 3rd place awards being given out. In fact, we want to mail out the awards this week — but the problem is that all we have is a single database with two tables, Posts and Students.

   Here is a portion of the Posts table:

	PostID	StudentID	GoodPost
1	1	1	TRUE
2	2	1	FALSE
3	3	2	TRUE
4	4	1	FALSE
5	5	2	TRUE
6	6	3	TRUE
7	7	3	TRUE
8	8	3	TRUE
9	9	2	FALSE
10	10	4	FALSE

   The third column, GoodPost, records whether the Piazza post was good as judged by your TAs.

   And here is a portion of the Students table:

	StudentID	StudentName	Address
1	1	John	123 Sesame St
2	2	Sophie	101 Ithaca Rd
3	3	Pierre	92 College Ave
4	4	Maria	24 Lake St

   We want to determine where to mail each of the awards, using SQL queries.

(a) (5 points) Create a view, GoodPostsCount, that shows the number of good posts each student has made. It should have two columns:

- StudentID
- NumGoodPosts, which records the number of good posts made by the student.

For example, the table might look like this:

	StudentID	NumGoodPosts
1	3	3
2	2	2
3	1	1

(Order of rows does not matter.)

Solution:

```
CREATE VIEW GoodPostsCount AS
SELECT StudentID, COUNT(*) AS NumGoodPosts
FROM Posts
WHERE Posts.GoodPost = "TRUE"
GROUP BY StudentID
ORDER BY NumGoodPosts DESC
```

The "CREATE VIEW" part is optional, since you could have used the GUI instead.

(b) (5 points) Create a view, PairwiseComparisons, that has one row for each pair of student IDs, and four columns:

- PostedMoreID is the StudentID of the student who posted more good piazza posts
- PostedLessID is the StudentID of the student who posted fewer good piazza posts
- MorePosts is the number of good posts made by PostedMoreID
- LessPosts is the number of good posts made by PostedLessID

Your result should look like this:

	PostedMoreID	MorePosts	PostedLessID	FewerGoodPosts
1	3	3	3	3
2	3	3	2	2
3	3	3	1	1
4	2	2	2	2
5	2	2	1	1
6	1	1	1	1

Notice that there is one row comparing each student with themselves. You may assume that there are no ties: no two students have made exactly the same number of good piazza posts.

Solution:

```
CREATE VIEW PairwiseComparisons AS
SELECT T1.StudentID AS PostedMoreID, T1.NumGoodPosts AS MorePosts,
T2.StudentID AS PostedLessID, T2.NumGoodPosts AS FewerGoodPosts
FROM GoodPostsCount AS T1, GoodPostsCount AS T2
WHERE T1.NumGoodPosts >= T2.NumGoodPosts
```

The "CREATE VIEW" part is optional, since you could have used the GUI instead.

(c) (5 points) Using the results of the previous view, create a table with three columns:

- StudentID
- Address
- Rank: the student with the most good posts should have value 1, the student with the second most good posts should have value 2, etc.

and three rows corresponding to the top three ranked students, ordered by rank. Your result should look like this:

	StudentID	Address	Rank
1	3	92 College Ave	1
2	2	101 Ithaca Rd	2
3	1	123 Sesame St	3

Solution:

```
SELECT PostedLessID AS StudentID, Address, COUNT(*) AS Rank
FROM PairwiseComparisons, Students
WHERE PairwiseComparisons.PostedLessID=Students.StudentID
GROUP BY StudentID
ORDER BY Rank ASC
LIMIT 3
```

The "CREATE VIEW" part is optional, since you could have used the GUI instead.

4. *PearPhone.* You lead a large tech company that produces phones called the PearPhone. Fortunately, the market is quite predictable and you have a constant demand rate over time. This demand rate is 18 per month. Suppose you need to reorder chips to produce your phone and the lead time is 1 week. Here, we will operate under EOQ conditions and we let assembly cost be virtually \$0 because it will be built by robots in a very quick time efficient manner. The market price of each phone sold is \$500 whereas, the chips required to build the phone cost \$100 and the fixed cost of a shipment of chips is \$100. Finally, holding chips required for a single PearPhone in your warehouse costs \$1 per week. For this question, you may assume every month is 4 weeks.

   Using this information, find the following.

   (a) (3 points) What is the optimal order quantity?

   Solution: We use the EOQ formula $Q^* = \sqrt{\frac{2K\lambda}{h}}$. The fixed cost per order is $K = \$100$, the monthly demand rate is $\lambda = 18$ chips, and the monthly holding cost is 4 times the weekly holding cost of \$1/chip, which is \$4/chip. This gives an optimal order quantity of:

$$Q^* = \sqrt{\frac{2(100)(18)}{4}} = \sqrt{900} = 30 \text{ chips}$$

   Note that using weekly values for both λ and h gives the same result.

   (b) (3 points) What is the average profit made every month?

   Solution: Monthly revenue is $\frac{\$500}{\text{phone}} \times \frac{18 \text{ phones}}{\text{month}} = \$9000$ per month. Monthly costs are computed using the formula for $G(Q)$ derived in class.

$$G(Q) = \frac{K\lambda}{Q} + c\lambda + \frac{hQ}{2}$$

   Note that λ and h are in monthly units so that $G(Q)$ represents the monthly cost.

$$G(Q^* = 30) = \frac{(100)(18)}{30} + (100)(18) + \frac{(4)(30)}{2} = 60 + 1800 + 60 = \$1920$$

   Monthly profit equals monthly revenue minus monthly costs, so

$$\text{Monthly Profit} = \$9000 - \$1920 = \$7080$$

(c) (3 points) What is the inventory level when an order is made?

Solution: Replenishment orders are placed when there is just enough inventory on hand to fulfil demand until the replenishment arrives, which takes one lead time τ. The reorder point R is therefore

$$R = \lambda\tau = \frac{9}{2} \text{ chips/week} \times 1 \text{ week} = 4.5 \text{ chips}$$

Note that it is impossible to place a replenishment order exactly when there are 4.5 chips left because chips are only used in multiples of 1. Therefore, the most reasonable reorder point is 5, because it avoids stockouts.

(d) (3 points) How many orders will be placed in one year?

Solution: The time between each order is T^*, computed below:

$$T^* = \frac{Q^*}{\lambda} = \frac{30 \text{ chips}}{18 \text{ chips/month}} = \frac{10}{6} \text{ months}$$

There are 12 months per year, so $\frac{12}{(10/6)} = 7.2$ orders are placed per year. Also note that the time between each order can be calculated using the weekly demand rate, which gives:

$$T^* = \frac{Q^*}{\lambda} = \frac{30 \text{ chips}}{4.5 \text{ chips/week}} = \frac{20}{3} \text{ weeks}$$

Assuming 52 weeks per year, this gives a slightly different answer of $\frac{52}{(20/3)} = 7.8$ orders placed per year.

5. *Newsvendor.* Consider the Newsvendor Problem, where we are selling newspapers, which we buy for \$1 and sell for \$1.50. If we have more newspapers than the demand the remaining newspapers will be recycled, and we get \$0 for each of these.

   Assume that the demand is a random variable with cdf $F(\cdot)$. Assume that this cdf is invertible, and denote this inverse by $F^{-1}(\cdot)$.

   We modeled the cost of newsvendor by thinking about the "opportunity costs". For each newspaper that could have been sold, the newsvendor is missing out on some potential profit (this was the "underage cost"), and for each newspaper that was not sold, the newsvendor is paying a cost (this was the "overage cost").

   (a) (3 points) Give the value of the per unit overage cost and the per unit underage cost for the example above.

   Solution: Overage cost represents the per unit loss incurred when demand is less than the order quantity. Underage cost represents the forgone per unit profit that we could have made by ordering more when demand is greater than the order quantity. Their values are calculated below:

$$c_o = \$1.00 - \$0.00 = \$1.00$$
$$c_u = \$1.50 - \$1.00 = \$0.50$$

   (b) (3 points) Give an expression for the total cost if the demand is x (a number, not a random variable), and the number of newspapers at the start of the day is Q.

   Solution: Using the cost function derived in class:

$$\begin{aligned} G(Q,x) &= c_o(\text{units of overage}) + c_u(\text{units of underage}) \\ &= c_o(Q-x)^+ + c_u(x-Q)^+ \\ &= (1)(Q-x)^+ + (0.5)(x-Q)^+ \end{aligned}$$

(c) (3 points) Give an expression for the number of newspapers to buy that minimizes the expected cost.

Solution: Using the formula for Q^* derived in class:

$$Q^* = F^{-1}\left(\frac{c_u}{c_u + c_o}\right) = F^{-1}\left(\frac{0.5}{0.5+1}\right) = F^{-1}\left(\frac{1}{3}\right)$$

We will now consider the "Fish Fragrant Newsvendor Problem": In this problem, we are selling newspapers, which we buy for \$1 and sell for \$1.50. Now however, when we have 20 or less left at the end of the day, we can sell them to the local fish restaurant as wrappers for their fish dinners for \$1 per newspaper. If we have more left, we can sell 20 to the local fish restaurant for \$1 per newspaper, but have to bring the remaining newspapers to the recycling, and we get \$0 for each of those.

We again want to decide the optimal number of newspapers to buy at the start of the day, Q. As in the normal Newsvendor Problem, we will think about minimizing the opportunity cost.

(d) (3 points) Give an expression for the cost when we decide to start off with Q newspapers and the demand is x (a number, not a random variable) — we'll refer to this as $G(Q, x)$.

Solution: Note that in this situation, we can order up to 20 extra newspapers (beyond the number demanded) without incurring any overage cost because we can sell the first 20 extras at the same price they are purchased at. We incur overage costs when we order 21 or more newspapers beyond demand, i.e. when $Q > x + 20$, or equivalently, when $Q - 20 - x$ is positive.

$$\begin{aligned} G(Q,x) &= c_o(\text{units of overage}) + c_u(\text{units of underage}) \\ &= c_o(Q-20-x)^+ + c_u(x-Q)^+ \\ &= (Q-20-x)^+ + (0.5)(x-Q)^+ \end{aligned}$$

Note that there is no effect on the underage cost incurred because ordering too few papers results in the same per unit forgone profit as before.

(Hint: it may help you to answer these questions, to derive the expression. These answers will not be graded.

1. Suppose x is the demand for newspapers in a day (a number, not a random variable) and suppose $x \geq Q + 1$. How much more profit could we have made

if we decided to buy $Q+1$ newspapers at the start of the day, instead of just Q newspapers?

2. Suppose x is the demand for newspapers in a day (a number, not a random variable) and suppose $Q \geq x+20$. What would have happened to our profit if we decided to buy $Q+1$ newspapers at the start of the day, instead of just Q newspapers?
3. Suppose x is the demand for newspapers in a day (a number, not a random variable) and suppose $x \leq Q \leq x+19$ (in other words, we have ordered at least as many newspapers as the demand, and there are 19 or fewer newspapers left at the end of the day). What would have happened to our profit if we decided to buy $Q+1$ newspapers at the start of the day, instead of just Q newspapers?)

(e) (3 points) Find the derivative of the expression $G(Q,x)$ with respect to Q.

Solution:

$$\begin{aligned}\frac{d}{dQ}G(Q,x) &= \frac{d}{dQ}\big[c_o(Q-20-x)^+ + c_u(x-Q)^+\big] \\ &= \frac{d}{dQ}\big[c_o(Q-20-x)^+\big] + \frac{d}{dQ}\big[c_u(x-Q)^+\big] \\ &= c_o\mathbb{1}_{\{Q>x+20\}} - c_u\mathbb{1}_{\{Q<x\}} \\ &= \mathbb{1}_{\{Q>x+20\}} - (0.5)\mathbb{1}_{\{Q<x\}}\end{aligned}$$

If you did not get an expression for the cost, you can use $\frac{d}{dQ}G(Q,x) = 3\mathbb{1}_{\{(Q-x)^+\}} + 5\mathbb{1}_{\{(x-Q-10)^+\}}$ from here on, where $\mathbb{1}_{\{\cdot\}}$ is the indicator function as you have seen in class.

(f) (3 points) Now let D be a random variable denoting the demand. Give an expression for the expected cost when we decide to start the day with Q newspapers.

Solution: We take the expectation of the cost function found in part (d).

$$\begin{aligned}G(Q,D) &= c_o(Q-20-D)^+ + c_u(D-Q)^+ \\ \mathbb{E}[G(Q,D)] &= \mathbb{E}[c_o(Q-20-D)^+ + c_u(D-Q)^+] \\ &= \mathbb{E}[c_o(Q-20-D)^+] + \mathbb{E}[c_u(D-Q)^+] \\ &= c_o\mathbb{E}[(Q-20-D)^+] + c_u\mathbb{E}[(D-Q)^+] \\ &= \mathbb{E}[(Q-20-D)^+] + (0.5)\mathbb{E}[(D-Q)^+]\end{aligned}$$

(g) (3 points) Take the derivative of the expected cost with respect to Q assuming that the derivative of this expectation is equal to the expectation of the derivative that you found in e. Simplify the expression as much as possible.

Solution:

$$\begin{aligned}
\frac{d}{dQ}\mathbb{E}[G(Q,D)] &= \frac{d}{dQ}\Big[c_o\mathbb{E}[(Q-20-D)^+] + c_u\mathbb{E}[(D-Q)^+]\Big] \\
&= c_o\mathbb{E}\left[\frac{d}{dQ}(Q-20-D)^+\right] + c_u\mathbb{E}\left[\frac{d}{dQ}(D-Q)^+\right] \\
&= c_o\mathbb{E}[\mathbb{1}_{\{Q>D+20\}}] - c_u\mathbb{E}[\mathbb{1}_{\{Q<D\}}] \\
&= c_o\mathbb{E}[\mathbb{1}_{\{D<Q-20\}}] - c_u\mathbb{E}[\mathbb{1}_{\{D>Q\}}] \\
&= c_o\mathbb{P}[D<Q-20] - c_u\mathbb{P}[D>Q] \\
&= c_oF(Q-20) - c_u(1-F(Q)) \\
&= F(Q-20) - (0.5)(1-F(Q))
\end{aligned}$$

(h) (3 points) Find an expression that must be true for the optimal number of newspapers that we should buy at the start of the day based on the previous question. If you did not find a derivative in the previous part, you can pretend that the derivative of the expectation equals $F(Q) - 2(1 - F(Q+3))$, where $F(\cdot)$ is the cdf of the random variable D, and use this in this question.

Solution: The minimum cost occurs when the derivative of the expected cost function equals zero (i.e. $\frac{d}{dQ}\mathbb{E}[G(Q,D)] = 0$). Therefore, any rearrangement of the following expression must be true.

$$F(Q-20) - (0.5)(1-F(Q)) = 0$$

(i) (3 points) Explain why you can use bisection search to find the optimal number of newspapers (approximately).

Solution: Rewriting the equation found in part (h) gives:

$$F(Q-20) + (0.5)F(Q) = 0.5$$

The left hand side, $F(Q-20) + (0.5)F(Q)$, is the sum of two monotonically increasing functions of Q, so it is also monotonically increasing in Q. Therefore, a bisection search is guaranteed to converge to Q^* as long as the initial bounds contain this value.

Question	Points	Score
1	10	
2	15	
3	15	
4	12	
5	27	
Total:	79	