Worksheet 6

This worksheet is due Monday of next week. You are encouraged to work in groups of up to 3 total students, but each student should make their own submission on Canvas. (It's fine for everyone in the group to have the same submission.)

Put the **full names** of everyone in your group (even if you're working alone) here. (This makes grading easier.)

- **Names**: Ilyas

In [1]:
```
import pandas as pd
import altair as alt
import numpy as np
```

- Import the attached "Math2B_grades_clean.csv" file, and name the DataFrame `df`.

In [2]:
```
df = pd.read_csv("../Data/Math2B_grades_clean.csv")
df.head()
```

Out[2]:

	Student_id	Quiz 1	Quiz 2	Midterm 1	Quiz 3	Quiz 4	Midterm 2	Quiz 5	Final exam	Webwork	Total
0	38649	70	30	58	50	70	44	60	26	39	F
1	10732	70	100	86	100	100	82	90	68	97	B
2	91531	80	70	64	90	90	84	80	63	89	C
3	61384	100	100	94	100	100	94	90	90	100	A
4	23583	80	80	84	100	90	92	70	73	99	B

- Using Boolean indexing, find the sub-DataFrame where the course grade ("Total") is "F" and where the Midterm 2 score is strictly greater than 72. Name this sub-DataFrame `df_sub`.

In [3]:
```
df_sub = df[(df["Total"] == "F") & (df["Midterm 2"] > 72)]
df_sub.head()
```

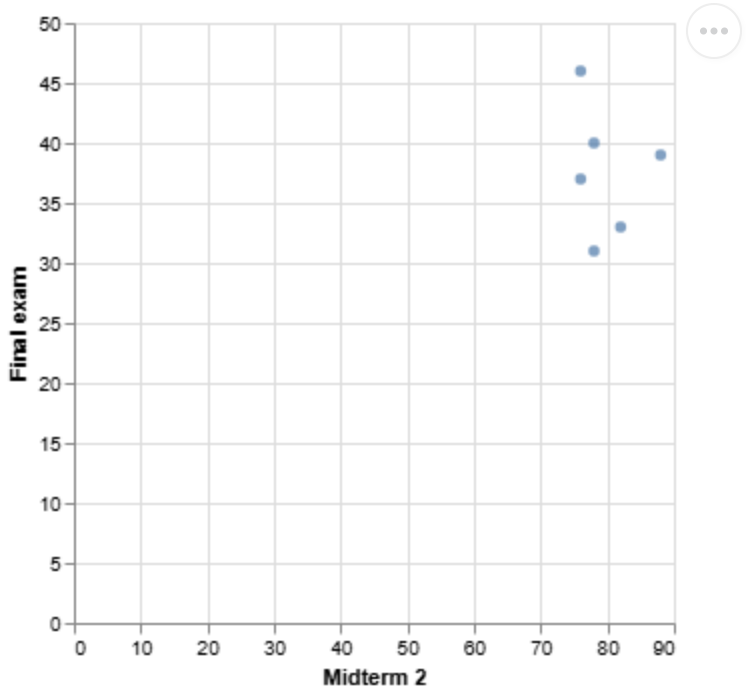
Out[3]:

	Student_id	Quiz 1	Quiz 2	Midterm 1	Quiz 3	Quiz 4	Midterm 2	Quiz 5	Final exam	Webwork	Tota
63	93619	0	40	68	100	80	88	80	39	43	
122	63723	0	70	60	0	90	76	60	46	32	
150	70693	80	70	64	100	80	76	30	37	46	
154	93643	60	40	50	60	50	78	70	31	91	
200	81308	80	50	52	80	70	78	0	40	74	

- Using Altair, make a scatter plot using the data from `df_sub` for which the x-coordinate is "Midterm 2" and the y-coordinate is "Final exam".

In [4]:
```
alt.Chart(df_sub).mark_circle().encode(x = "Midterm 2", y = "Final exam")
```

Out[4]:



- Based on the chart, how many rows do you expect are in `df_sub`? Explain your answer in a markdown cell, and check your answer using pandas.

(This approach isn't guaranteed to be correct, because hypothetically two points might be on top of each other, or a row could contain missing data.)

As there are 6 points, we can expect 6 rows in `df_sub`

In [5]:
```
print(f"There are {df_sub.shape[0]} rows in `df_sub`")
```

```
There are 6 rows in `df_sub`
```

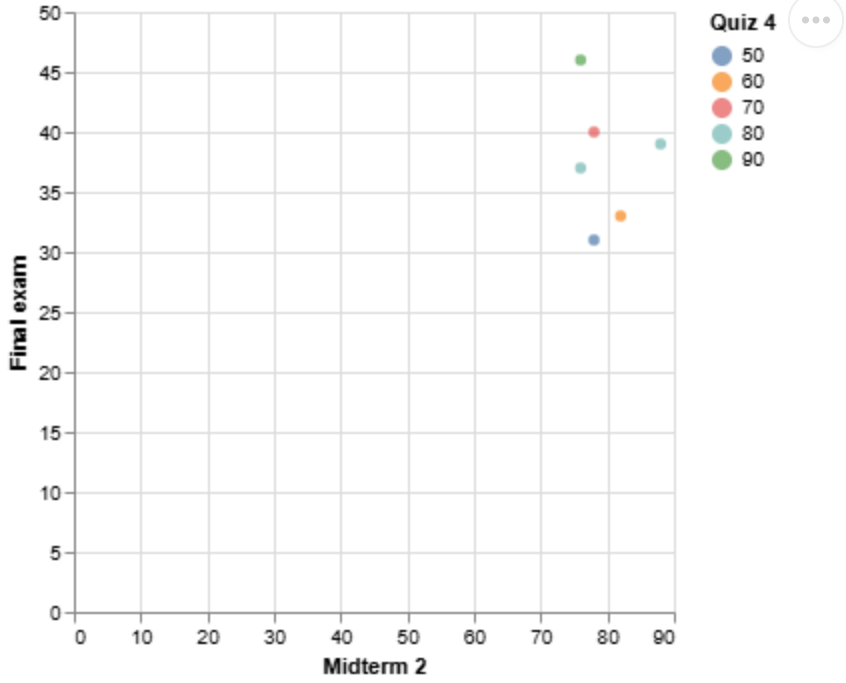
Add one or more additional visual channels (don't use `tooltip` here) to the chart (but not changing the `x` or `y` definitions) so that you can tell which of these students had the lowest score on Quiz 4.

Some options:

- `color` (you might want to use the encoding data type `:O` or `:N` to make the colors more distinct Reference).
- `size`
- If you change to `mark_point`, you can use the `shape` visual channel. I don't think `shape` works with quantitative data, so you need to use an encoding data type like `:N` in this case. I personally prefer using `mark_point(filled=True)` over `mark_point()`.

In [6]:
```
alt.Chart(df_sub).mark_circle().encode(x = "Midterm 2", y = "Final exam", color="Qu
```

Out[6]:



- Explain in a markdown cell how you can tell from the chart which point has the lowest score on Quiz 4.

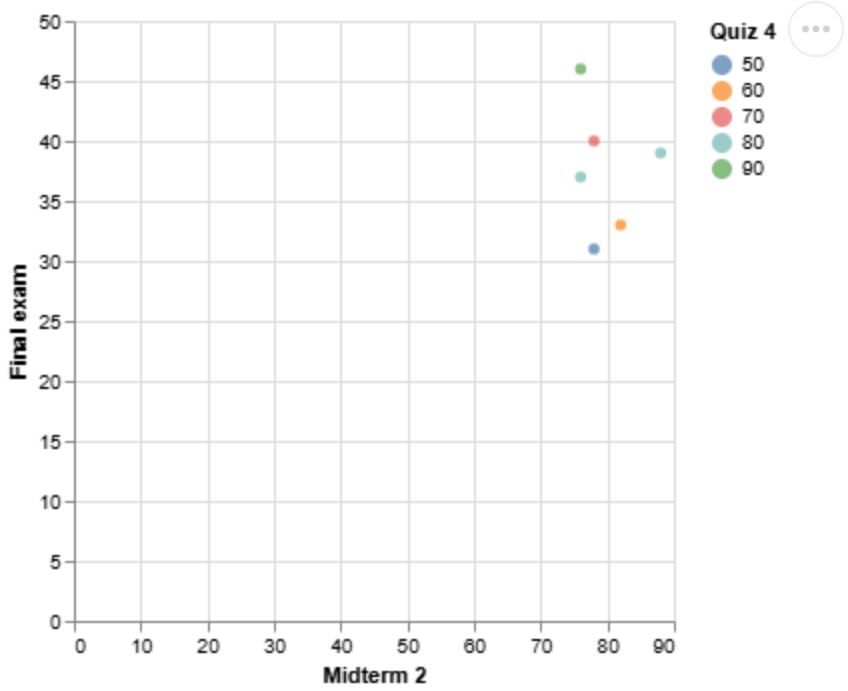
Using the color encoding of quiz 4 scores in the points, we can see that the lowest point has the lowest quiz 4 score

- Add a tooltip with "Student_id" so that you can find the student id of the corresponding student.

Your code for this question will also be used at the end of this worksheet.

In [7]:
```
alt.Chart(df_sub).mark_circle().encode(x = "Midterm 2", y = "Final exam", color="Qu
```

Out[7]:



- Here is a way to find that same student id using pandas. Can you figure out how the following code works by breaking it up into pieces? (There might be a question based on this code on the next quiz or on the midterm.)

  `df_sub.set_index("Student_id")["Quiz 4"].idxmin()`

In [8]:
```
df_index = df_sub.set_index("Student_id")
df_quiz = df_index["Quiz 4"]
index = df_quiz.idxmin()
print(index)
```

```
93643
```

This is first setting the index to be the student id. Secondly extracting the quiz 4 column. Finally getting the index of the minimum value, which is now the student id by step 1.

- Why does the following code give a different answer? (Hint. Display `df_sub`.)

  `df_sub["Quiz 4"].idxmin()`

In [9]:
```
df_sub.head()
```

Out[9]:

	Student_id	Quiz 1	Quiz 2	Midterm 1	Quiz 3	Quiz 4	Midterm 2	Quiz 5	Final exam	Webwork	Tota
63	93619	0	40	68	100	80	88	80	39	43	
122	63723	0	70	60	0	90	76	60	46	32	
150	70693	80	70	64	100	80	76	30	37	46	
154	93643	60	40	50	60	50	78	70	31	91	
200	81308	80	50	52	80	70	78	0	40	74	

This is getting the row in df_sub, which is still labedled using the row numbers from df, not using the student id as the index

- What changes if we use `argmin` instead of `idxmin`? What is the difference between these two methods? How does this correspond to what you see in `df_sub`? Answer in a markdown cell.

Using `argmin` gives us the deafault index going from 0 - len(df) - 1. Using `idxmin` gives us the index from the index column in the data frame.

- If you were to encode the "Student_id" value in one of these channels, why would `"Student_id:N"` make much more sense than `"Student_id:Q"` or `"Student_id:O"`? Explain in a markdown cell.

Since student ids are discrete not continuous, using `Student_id:Q` doesn't really make sense. In addition, having different colors makes it easier to differentiate students, especially in the case where two students overlap. Therefore `Student_id:N` is a better choice than `Student_id:O`.

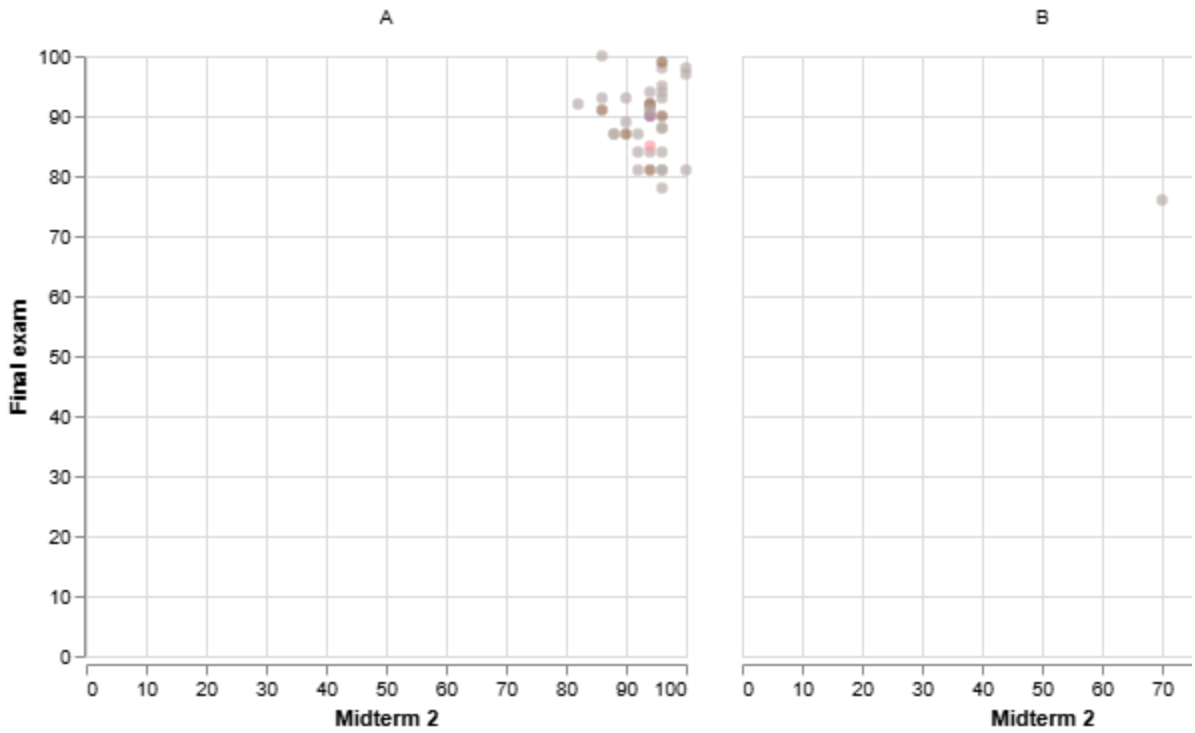
Take your same Altair chart code above (the one where you found the student id using the `tooltip`) and make the following changes to it.

- Change from `df_sub` to the full DataFrame `df`.
- Add `column="Total"` to the encoding.

- For the student whose student id you found above, where is the corresponding point located in this facet chart? Explain in a markdown cell where is this point and how you can tell. (You should be able to check that you are correct using the tooltip.)

In [10]:
```
alt.Chart(df).mark_circle().encode(x = "Midterm 2", y = "Final exam", color="Quiz 4
```

Out[10]:



We can tell where the student is using 3 factors.

1. The student should be in the "F" category as that was one of the filters we used when making `df_sub`.
2. The student should have a midterm 2 score > 72 as that was the other filter for making `df_sub`.
3. The `color` paramter shows us the students who got a 50 on quiz 4 in green, which allows us to find the studnt.

Submission

- Reminder: everyone needs to make a submission on Canvas.
- Reminder: include everyone's full name at the top, after **Names**.
- Using the `Share` button at the top right, enable public sharing, and enable Comment privileges. Then submit the created link on Canvas.