

Applying Slip Days while Maximizing Student Grades

`101_final_grades.ipynb

Slip Days

Calculate Total Late Days & Allocable Slip Days

```
1 late, slip = 0, 0
2 for n in range(1, 5):
3     late += roster[f'Project {n} Lateness']
4     slip += roster[f'Project {n} Traded-in Slip Days']
5
6 for n in range(1, 6):
7     late += roster[f'Multivitamin {n} Lateness']
8     slip += roster[f'Multivitamin {n} Traded-in Slip Days']
9
10 roster['Total Lateness'] = late
11 roster['Total Slip Days Remaining'] = 9 - slip
```

[174]

- First step is to check how many total "late" days a student has accumulated
 - 9 is the total number of slip days available to each student
 - Some students had previous extensions/extenuating circumstances so they've used some of their slip days. This is accounted for by `slip`

Determine how to apply slip days for students who have more late days than remaining slip days

```
1 to_be_updated = list(roster[roster['Total Lateness'] > roster['Total Slip Days Remaining']].index)
2 to_be_updated
```

[5]

- `to_be_updated` contains a list of student emails who have more late days than remaining slip days. **We need to figure out how to maximize grades for these students.**

```
1 to_process = {}
2 for email in to_be_updated:
3     student_late_items = []
4     for item in lateness_items:
5         slip_day_curr = roster.loc[email, item + ' Lateness']
6         if slip_day_curr > 0:
7             slip_day_curr = int(slip_day_curr)
8             student_late_items.append(item)
9     student_late_items.sort(key=lambda item: roster.loc[email, item + ' Score'], reverse=True)
10    print(email, student_late_items)
11    to_process[email] = student_late_items
```

[178]

- This cell prints out the late assignment items for each student in the `to_be_updated` list. Useful for debugging/verifying if slip days are correctly applied.
- `to_process` is a dictionary with key `email: string` and value `student_late_items: List`
- Note that `student_late_items` is sorted by score from highest to lowest. This makes applying slip days easier - we just need to try applying slip days to assignments where students scored the highest points.
 - e.g. If a student scored 0/25 in their multivitamin (late = 1), then we won't bother applying slip days to this assignment, because they'll be better off if we applied this slip day to their late project instead.

```

1 for email in to_process:
2     print(f'Start processing {email}:')
3     items = to_process[email]
4     print(f'Late assignments: {items}')
5     items_to_remove = []
6     for item in items:
7         lateness_curr = roster.loc[email, item + ' Lateness']
8         slip_days_remaining = roster.loc[email, 'Total Slip Days Remaining']
9         if slip_days_remaining >= lateness_curr:
10            roster.loc[email, item + ' Auto-Allocated Slip Days'] += lateness_curr
11            roster.loc[email, 'Total Slip Days Remaining'] -= lateness_curr
12            items_to_remove.append(item)
13            print(f'No penalty on {item}')
14
15
16        else:
17            while slip_days_remaining > 0:
18                roster.loc[email, item + ' Auto-Allocated Slip Days'] += 1
19                roster.loc[email, 'Total Slip Days Remaining'] -= 1
20                slip_days_remaining -= 1
21
22    for item in items_to_remove:
23        items.remove(item)
24    print(f'Remaining: {items}')
25    for item in items:
26        penalty_days = roster.loc[email, item + ' Lateness'] - roster.loc[email, item + ' Auto-Allocated Slip Days']
27        roster.loc[email, item + ' Late Penalty'] = penalty_days * 15.0

```

- This is the cell for actually applying slip days while maximizing student grades. The logic is as follows:
 - If the student has enough slip days to cover one of their late assignments, then apply the slip days. Eventually we remove the assignment from the list so no late penalty is applied.
 - Otherwise, the student has fewer slip days than the number of late days on the given assignment. So here we are just on a "best-effort basis" - apply the remaining slip days to that assignment.