Homework 2

Byteflow Dynamics

10/1/2017

Grade Calculator

Using the same dataset, calculate percent grades and assign a letter grade to each student based on the grading policy below.

- Midterm, Final, Extra credit scores are on a 100 scale, all other scores are on a 10 scale (Extra credit used to supplement exam scores)
- Final grade is based 25% on Midterm and Final exam each, 20% on Homework, 10% on Quizzes, and 20% on lab (No need to calculate lecture grade separately)
- 3 lowest lab scores are dropped, other labs all weight equally
- Extra credit for lecture is added to the final exam score on a 100 scale
- Letter grades are determined based on following grading scale (You don't have to use quantiles!) :
  - Above 95: A+
  - Between 90-95: A
  - Between 85-90: B+
  - Between 80-85: B
  - Between 70-80: C
  - Between 40-70: D
  - Below 40: F

```r
scores_exam <- scores %>%
  group_by(Last.Name, First.Name) %>%
  filter(type == "Midterm" | type == "Final.exam" | type == "Extra.Credit") %>%
  summarise(exam_weighted = sum(score) * 0.25) # 25% weight on midterm and final
scores_exam
```

```
## # A tibble: 27 x 3
## # Groups:   Last.Name [?]
##    Last.Name First.Name exam_weighted
##        <chr>      <chr>         <dbl>
##   1         A          T         25.25
##   2         B          B         41.75
##   3         B          J         49.75
##   4         B          M          0.00
##   5         C          L         35.25
##   6         C          M         37.00
##   7         C          S         41.00
##   8         C          T         52.25
##   9         D          C         44.25
## 10         D          R         27.75
## # ... with 17 more rows
```

```r
scores_hw <- scores %>%
  group_by(Last.Name, First.Name) %>%
  filter(grade_type == "HW") %>%
  summarise(hw_weighted = mean(score) * 2) #multiply by 2 to make it out of 20
scores_hw
```

```
## # A tibble: 27 x 3
## # Groups:   Last.Name [?]
##    Last.Name First.Name hw_weighted
##        <chr>      <chr>       <dbl>
##  1         A          T         9.2
##  2         B          B        12.6
##  3         B          J        16.8
##  4         B          M         0.0
##  5         C          L        13.4
##  6         C          M        14.4
##  7         C          S        16.8
##  8         C          T        10.8
##  9         D          C        13.2
## 10         D          R        17.4
## # ... with 17 more rows
```

```
scores_quiz <- scores %>%
  group_by(Last.Name, First.Name) %>%
  filter(grade_type == "Quiz") %>%
  summarise(quiz_weighted = mean(score)) # quiz scores are already out of 10 so no need to multiply wit
scores_quiz
```

```
## # A tibble: 27 x 3
## # Groups:   Last.Name [?]
##    Last.Name First.Name quiz_weighted
##        <chr>      <chr>         <dbl>
##  1         A          T          6.75
##  2         B          B          7.40
##  3         B          J          8.90
##  4         B          M          0.00
##  5         C          L          5.75
##  6         C          M          8.50
##  7         C          S          7.15
##  8         C          T          7.75
##  9         D          C          6.25
## 10         D          R          4.75
## # ... with 17 more rows
```

```
lab_ranked <- scores %>%
  group_by(Last.Name, First.Name) %>%
  filter(grade_type == "Lab") %>%
  mutate(rank = row_number(score)) %>%
  arrange(Last.Name, First.Name, rank) # to check the ranking function is working

lab_dropped <- lab_ranked %>%
  filter(rank > 3)

scores_lab <- lab_dropped %>%
  summarise(lab_grade = mean(score) * 2)

scores_lab
```

```
## # A tibble: 27 x 3
## # Groups:   Last.Name [?]
##    Last.Name First.Name lab_grade
##        <chr>      <chr>     <dbl>
##  1         A          T  7.818182
##  2         B          B 19.818182
##  3         B          J 20.000000
##  4         B          M  0.000000
##  5         C          L 19.818182
##  6         C          M 17.454545
##  7         C          S 19.090909
##  8         C          T 18.727273
##  9         D          C 19.818182
## 10         D          R 19.818182
## # ... with 17 more rows
```

```
final_grade <- inner_join(scores_exam, scores_hw) %>%
  inner_join(scores_quiz) %>%
  inner_join(scores_lab) %>%
  mutate(grade_all = (exam_weighted + hw_weighted + quiz_weighted + lab_grade)) %>%
  select(Last.Name, First.Name, grade_all)
```

```
## Joining, by = c("Last.Name", "First.Name")
## Joining, by = c("Last.Name", "First.Name")
## Joining, by = c("Last.Name", "First.Name")
```

```
final_grade
```

```
## # A tibble: 27 x 3
## # Groups:   Last.Name [14]
##    Last.Name First.Name grade_all
##        <chr>      <chr>     <dbl>
##  1         A          T  49.01818
##  2         B          B  81.56818
##  3         B          J  95.45000
##  4         B          M   0.00000
##  5         C          L  74.21818
##  6         C          M  77.35455
##  7         C          S  84.04091
##  8         C          T  89.52727
##  9         D          C  83.51818
## 10         D          R  69.71818
## # ... with 17 more rows
```

```
grade_Aplus <- final_grade %>%
  filter(grade_all >= 95) %>%
  mutate(grade = "A+")

grade_A <- final_grade %>%
  filter(grade_all >= 90 & grade_all < 95) %>%
  mutate(grade = "A")

grade_Bplus <- final_grade %>%
  filter(grade_all >= 85 & grade_all < 90) %>%
  mutate(grade = "B+")
```

```
grade_B <- final_grade %>%
  filter(grade_all >= 80 & grade_all < 85) %>%
  mutate(grade = "B")

grade_C <- final_grade %>%
  filter(grade_all >= 70 & grade_all < 80) %>%
  mutate(grade = "C")

grade_D <- final_grade %>%
  filter(grade_all >= 40 & grade_all < 70) %>%
  mutate(grade = "D")

grade_F <- final_grade %>%
  filter(grade_all < 40) %>%
  mutate(grade = "F")

letter_grade <- bind_rows(grade_A, grade_B) %>%
  bind_rows(grade_C) %>%
  bind_rows(grade_D) %>%
  bind_rows(grade_F) %>%
  bind_rows(grade_Aplus) %>%
  bind_rows(grade_Bplus) %>%
  arrange(Last.Name, First.Name)

letter_grade
```

```
## # A tibble: 27 x 4
## # Groups:   Last.Name [14]
##    Last.Name First.Name grade_all grade
##        <chr>      <chr>     <dbl> <chr>
##  1         A          T  49.01818     D
##  2         B          B  81.56818     B
##  3         B          J  95.45000    A+
##  4         B          M   0.00000     F
##  5         C          L  74.21818     C
##  6         C          M  77.35455     C
##  7         C          S  84.04091     B
##  8         C          T  89.52727    B+
##  9         D          C  83.51818     B
## 10         D          R  69.71818     D
## # ... with 17 more rows
```