SRL-use total error counts to predict learning outcomes

Joyce Zhang

2023-03-29

1. import data

1.1 import data from "hwXX-snapshot-summary.csv" for HW0-8 (except for HW1, HW2)

create dataframe df. Each row shows the following 4 measures for each snapahot (see example below) - c_error: 1: snapshot with a compiler error, 0: snapshot compiled - r_error: the number of runtime errors in a snapshot - t_failed: the number of tests that failed (includes tests that failed because of the runtime error) - t_success: the number of tests that ended with success

student	snapshot	c_error	r_error	t_failed	t_success
008a13042777e1aaca446a68fbc5b3877e6ed232	2020-09-05T21_56_27.640Z	1	0	0	0
008a13042777e1aaca446a68fbc5b3877e6ed232	2020-09-05T21_58_45.783Z	0	0	0	1
008a13042777e1aaca446a68fbc5b3877e6ed232	2020-09-05T22_02_19.551Z	0	0	1	0
008a13042777e1aaca446a68fbc5b3877e6ed232	2020-09-05T22_10_52.760Z	0	0	1	0
008a13042777e1aaca446a68fbc5b3877e6ed232	2020-09-05T22_13_03.239Z	0	0	1	0

agggregate df to student level

create dataframe 'df_student'. each row shows for each student and HW, the total number of snapshots captured (N_snaphot), total number of snapshots containing compiler errors (C_totalSnap), its percentage (C_perc), total number of snapshot containing runtime errors (R_totalSnap) and its percentages (R_perc), total number of runtime errors (a snapshot can have more than 1 runtime error)(R_totalError), and the average number of runtime errors (R_avgError). See the top 5 rows of df_student dataframe below

```
## `summarise()` has grouped output by 'student'. You can override using the
## `.groups` argument.
```

student	HW	N_snapshot	C_totalSnap	C_perc	R_totalSnap	R_pe
008a13042777e1aaca446a68fbc5b3877e6ed232	HW0	14	1	0.0714286	0	0.00000
008a13042777e1aaca446a68fbc5b3877e6ed232	HW3	11	4	0.3636364	1	0.09090
008a13042777e1aaca446a68fbc5b3877e6ed232	HW4	11	1	0.0909091	5	0.45454
008a13042777e1aaca446a68fbc5b3877e6ed232	HW5	18	0	0.0000000	3	0.16666
008a13042777e1aaca446a68fbc5b3877e6ed232	HW6	83	4	0.0481928	8	0.09638

1.2 import data from "hwXX-compiler-errors.csv"files for the same HW

similarly, we imported the compiler errors from the "hwXX-compiler-errors.csv" files. aggregated them to the student level, creating a dataframe (df_c_student). This dataframe the total number of compiler errors a student has made for each HW. see table below

```
## `summarise()` has grouped output by 'HW'. You can override using the `.groups`
## argument.
```

HW	student	C_totalError
HW0	0de822bddc1dd72a3230320f51b10b9f7a509869	1
HW0	0fc86b6a1afaad8cc203fac7e27c55097c310937	8
HW0	107e619e6f2a5a39e729d6ffaa91b43d2be99edf	1
HW0	4542f79f46c1e57daf32196b082986d08d6addda	1
HW0	4b8cafbb2f0acbe010290da14340378e9c684d75	3

combine df_student with df_c_student see the top 5 rows below.

student	HW	N_snapshot	C_totalSnap	C_perc	R_totalSnap	R_pe
008a13042777e1aaca446a68fbc5b3877e6ed232	HW0	14	1	0.0714286	0	0.00000
008a13042777e1aaca446a68fbc5b3877e6ed232	HW3	11	4	0.3636364	1	0.09090
008a13042777e1aaca446a68fbc5b3877e6ed232	HW4	11	1	0.0909091	5	0.45454
008a13042777e1aaca446a68fbc5b3877e6ed232	HW5	18	0	0.0000000	3	0.16666
008a13042777e1aaca446a68fbc5b3877e6ed232	HW6	83	4	0.0481928	8	0.09638

2 descriptive stats for error rate by HW

HW	N_student	avg_N_snapshot	avg_N_runtime	avg_N_compiler
HW0	279	12	0.0	0
HW3	295	18	3.0	16
HW4	281	9	0.0	4
HW5	281	20	48.0	6
HW6	279	77	72.0	20
HW7	280	18	18.0	6
HW8	278	27	20.5	12

3 use total number of runtime/compiler errors to predict HW performance

the following regression is used to estimate the relationship between the total number of errors and HW performance (rank). The results show that controlling for HW number, total number of runtime and compiler errors are negatively associated with the ranking in HW performance. number of compiler errors is more significantly as well as has higher impact than the numner of runtime errors.

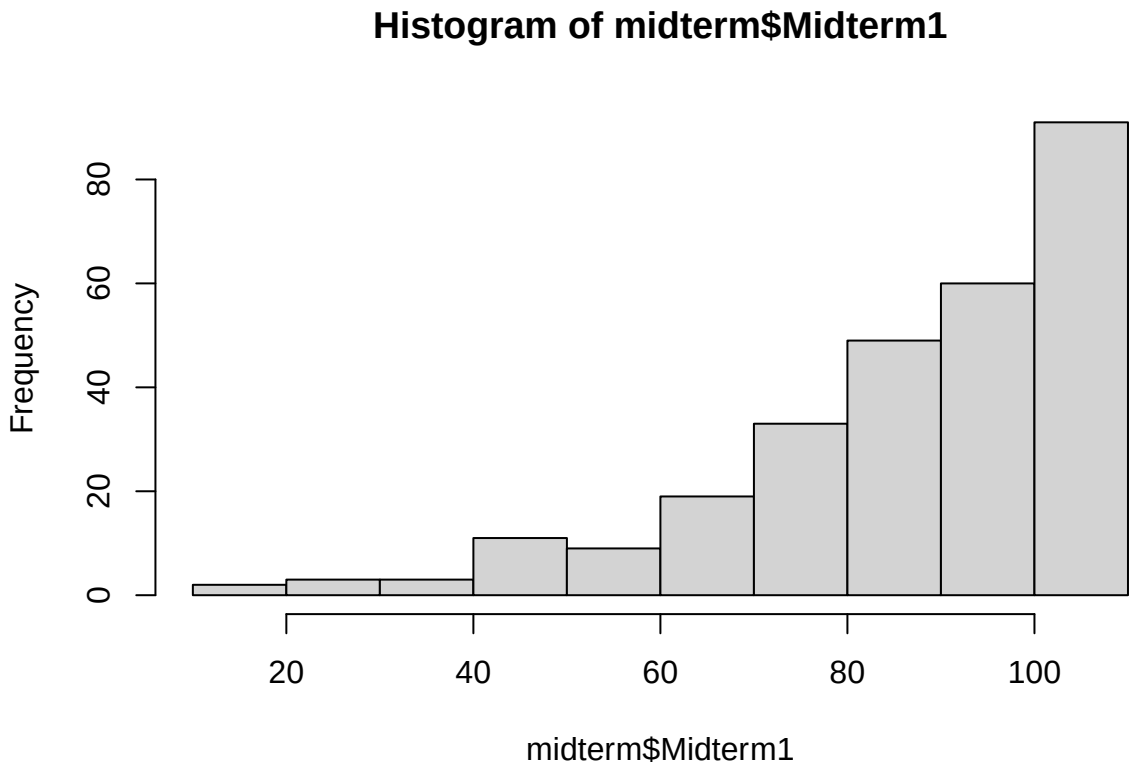
lm_HWgrade <- summary(lm(rank~HW+R_totalError+C_totalError, data = df_grade))

```
##
## Call:
## lm(formula = rank ~ HW + R_totalError + C_totalError, data = df_grade)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -154.28  -64.08   22.18   62.94  219.31
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  149.31663    4.47574  33.361  < 2e-16 ***
## HWHW3          4.35973    6.35051   0.687   0.4925
## HWHW4          6.58427    6.32202   1.041   0.2978
## HWHW5         12.64621    6.84169   1.848   0.0647 .
## HWHW6         18.32140    7.43541   2.464   0.0138 *
## HWHW7          9.79408    6.44674   1.519   0.1289
## HWHW8         12.77959    6.49190   1.969   0.0491 *
## R_totalError  -0.07100    0.03852  -1.843   0.0654 .
## C_totalError  -0.14529    0.03250  -4.470 8.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 73.95 on 1923 degrees of freedom
## Multiple R-squared:  0.01432,    Adjusted R-squared:  0.01022
## F-statistic: 3.492 on 8 and 1923 DF,  p-value: 0.0005183
```

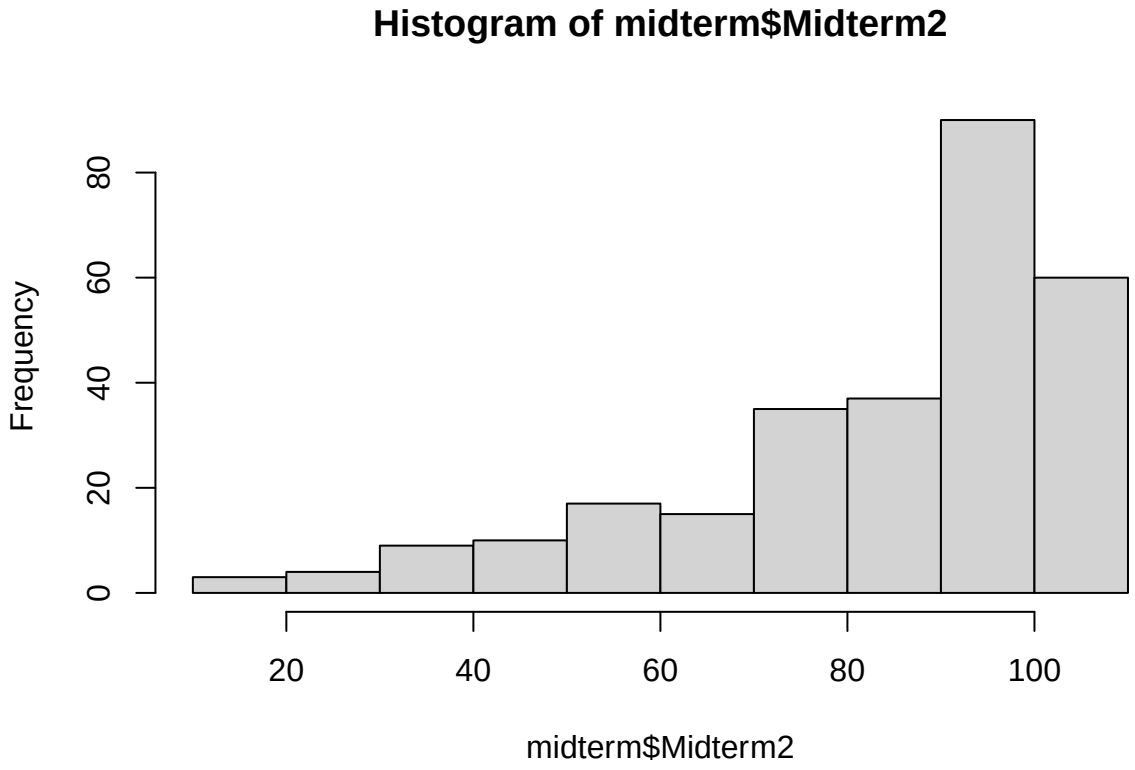
4 use totalt number of runtime/compiler errors to predict mid-term 1 & 2 grades

4.1 the distribution of midterm1 & 2

midterm 1 histogram (skewness = -1.24; kurtosis = 4.32)



midterm 2 histogram (skewness = -1.23; kurtosis = 3.83)



Because both grades not heavily skewed, in the following analyses we did not transform the grades into ranks.

4.2 predicting mid-term 1 grades

I tried two approaches to predict mid-term 1 grades (individual HW errors vs. aggregated errors for HW0-4)

```
# predict midterm1
# model 1
summary(lm(Midterm1 ~
             R_totalError_HW0+R_totalError_HW3+R_totalError_HW4+
             C_totalError_HW0+C_totalError_HW3+C_totalError_HW4,
           data = allError))
```

```
##
## Call:
## lm(formula = Midterm1 ~ R_totalError_HW0 + R_totalError_HW3 +
##     R_totalError_HW4 + C_totalError_HW0 + C_totalError_HW3 +
##     C_totalError_HW4, data = allError)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -74.850  -8.538   4.883  13.580  33.072
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      92.01830    1.56703  58.721  < 2e-16 ***
## R_totalError_HW0  8.12263   13.43999   0.604  0.54610
## R_totalError_HW3 -0.02722    0.15871  -0.172  0.86394
## R_totalError_HW4 -0.25977    0.18819  -1.380  0.16860
## C_totalError_HW0 -0.03165    0.26536  -0.119  0.90515
## C_totalError_HW3 -0.07041    0.02215  -3.178  0.00165 **
```

```
## C_totalError_HW4 -0.14568    0.05884  -2.476  0.01389 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.77 on 273 degrees of freedom
## Multiple R-squared:  0.1175, Adjusted R-squared:  0.09814
## F-statistic:  6.06 on 6 and 273 DF,  p-value: 5.677e-06
```

```
#model 2
summary(lm(Midterm1~totalRuntime_HW04+totalcompile_HW04,data = allError))
```

```
##
## Call:
## lm(formula = Midterm1 ~ totalRuntime_HW04 + totalcompile_HW04,
##     data = allError)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.789  -4.533   1.101   4.156  23.783
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.142386   2.305512   1.797   0.0735 .
## totalRuntime_HW04 0.483854   0.012306  39.317   <2e-16 ***
## totalcompile_HW04 0.003408   0.008035   0.424   0.6717
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.587 on 277 degrees of freedom
## Multiple R-squared:  0.8537, Adjusted R-squared:  0.8526
## F-statistic: 808.2 on 2 and 277 DF,  p-value: < 2.2e-16
```

4.3 predicting mid-term 2 grades

```
#model 1
summary(lm(Midterm2 ~
             R_totalError_HW0+R_totalError_HW3+R_totalError_HW4 +
             R_totalError_HW5+R_totalError_HW6+R_totalError_HW7+R_totalError_HW8+
             C_totalError_HW0+C_totalError_HW3+C_totalError_HW4+
             C_totalError_HW5+C_totalError_HW6+C_totalError_HW7++C_totalError_HW8,
           data = allError))
```

```
##
## Call:
## lm(formula = Midterm2 ~ R_totalError_HW0 + R_totalError_HW3 +
##     R_totalError_HW4 + R_totalError_HW5 + R_totalError_HW6 +
##     R_totalError_HW7 + R_totalError_HW8 + C_totalError_HW0 +
##     C_totalError_HW3 + C_totalError_HW4 + C_totalError_HW5 +
##     C_totalError_HW6 + C_totalError_HW7 + +C_totalError_HW8,
##     data = allError)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.534  -9.778   4.802  13.183  34.082
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      98.175107   2.419201  40.582  < 2e-16 ***
## R_totalError_HW0  9.419041  14.581715   0.646  0.51887
## R_totalError_HW3  0.022038   0.171555   0.128  0.89788
## R_totalError_HW4 -0.402799   0.199362  -2.020  0.04434 *
## R_totalError_HW5 -0.063400   0.019565  -3.241  0.00135 **
## R_totalError_HW6 -0.029371   0.014679  -2.001  0.04642 *
## R_totalError_HW7  0.032416   0.035055   0.925  0.35596
## R_totalError_HW8 -0.095676   0.040906  -2.339  0.02008 *
## C_totalError_HW0 -0.132514   0.278967  -0.475  0.63517
## C_totalError_HW3  0.014974   0.025141   0.596  0.55193
## C_totalError_HW4 -0.047209   0.062819  -0.752  0.45301
## C_totalError_HW5  0.006312   0.015371   0.411  0.68166
## C_totalError_HW6 -0.037900   0.022565  -1.680  0.09422 .
## C_totalError_HW7 -0.104795   0.049668  -2.110  0.03581 *
## C_totalError_HW8 -0.044615   0.019568  -2.280  0.02340 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.44 on 265 degrees of freedom
## Multiple R-squared:  0.1846, Adjusted R-squared:  0.1415
## F-statistic: 4.285 on 14 and 265 DF,  p-value: 7.774e-07
```

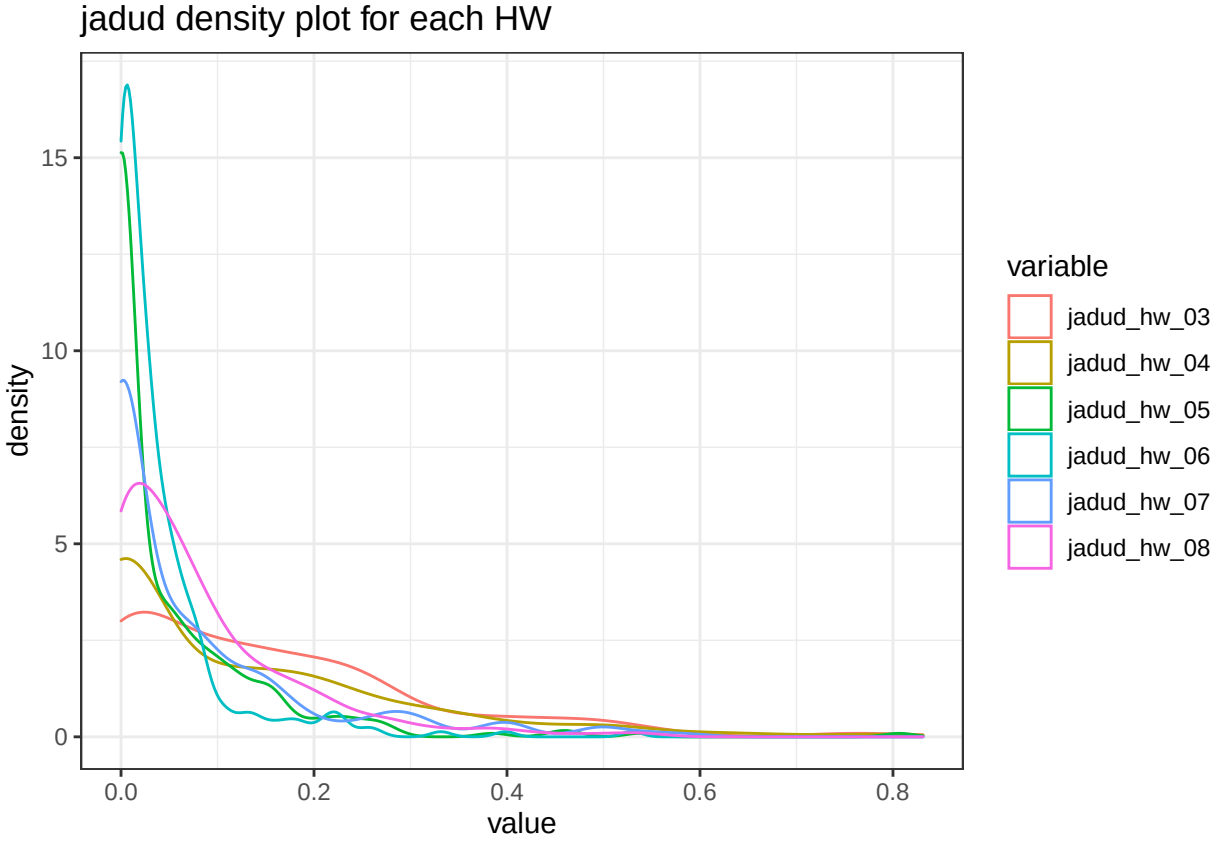
```
#model 2
summary(lm(Midterm2 ~ totalRuntime_HW08 + totalcompile_HW08, data = allError))
```

```
##
## Call:
## lm(formula = Midterm2 ~ totalRuntime_HW08 + totalcompile_HW08,
##     data = allError)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.268  -9.065   6.817  13.770  40.798
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       83.96498    4.11262  20.416  < 2e-16 ***
## totalRuntime_HW08  0.01535    0.01143   1.343     0.18
## totalcompile_HW08 -0.03226    0.00720  -4.481 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.32 on 277 degrees of freedom
## Multiple R-squared:  0.06895,    Adjusted R-squared:  0.06223
## F-statistic: 10.26 on 2 and 277 DF,  p-value: 5.046e-05
```

HW	Avg	Stdv
jadud_hw_03	0.15	0.16
jadud_hw_04	0.12	0.16
jadud_hw_05	0.05	0.09
jadud_hw_06	0.04	0.06
jadud_hw_07	0.08	0.12
jadud_hw_08	0.08	0.10

5 use jadud to predict mid-term 1 & 2 grades

5.1 descriptive stats and density plot of jadud error quotient



5.2 use jadud EQ to predict midterm 1 grades

```
# model 1
summary(lm(Midterm1 ~ jadud_hw_03 + jadud_hw_04, data = jadud_df))
```

```
##
## Call:
## lm(formula = Midterm1 ~ jadud_hw_03 + jadud_hw_04, data = jadud_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -67.90  -9.83   4.07  12.64  36.96
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   95.862      1.624  59.037  < 2e-16 ***
## jadud_hw_03  -54.915      7.799  -7.041 1.52e-11 ***
## jadud_hw_04   -5.674      7.015  -0.809    0.419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.05 on 275 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1712
## F-statistic:  29.6 on 2 and 275 DF,  p-value: 2.272e-12
```

```
# model 2
summary(lm(Midterm1 ~ Avg_HW04, data = jadud_df))
```

```
##
## Call:
## lm(formula = Midterm1 ~ Avg_HW04, data = jadud_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.107  -9.720   4.326  13.058  32.427
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   94.684      1.626  58.239  < 2e-16 ***
## Avg_HW04     -56.560      8.883  -6.368 7.91e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.5 on 278 degrees of freedom
## Multiple R-squared:  0.1273, Adjusted R-squared:  0.1241
## F-statistic: 40.55 on 1 and 278 DF,  p-value: 7.912e-10
```

5.2 use jadud EQ to predict midterm 2 grades

```
##
## Call:
## lm(formula = Midterm2 ~ jadud_hw_03 + jadud_hw_04 + jadud_hw_05 +
##     jadud_hw_06 + jadud_hw_07 + jadud_hw_08, data = jadud_df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -63.641  -8.749   3.690  12.007  39.815
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  93.8480     1.7491  53.656  < 2e-16 ***
## jadud_hw_03 -41.4734     9.1356  -4.540 8.54e-06 ***
## jadud_hw_04   6.0319     7.5889   0.795   0.4274
## jadud_hw_05  -0.3816    13.5468  -0.028   0.9775
## jadud_hw_06 -54.1715    20.9883  -2.581   0.0104 *
## jadud_hw_07 -17.2137    10.6318  -1.619   0.1066
## jadud_hw_08  -8.6400    13.4427  -0.643   0.5210
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.6 on 266 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.1929, Adjusted R-squared:  0.1746
## F-statistic: 10.59 on 6 and 266 DF,  p-value: 1.509e-10
```

```
##
## Call:
## lm(formula = Midterm2 ~ Avg_HW08, data = jadud_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -63.244  -8.684   4.702  12.630  41.468
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   92.948      1.722   53.98  < 2e-16 ***
## Avg_HW08     -78.429     11.078   -7.08 1.18e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.35 on 278 degrees of freedom
## Multiple R-squared:  0.1527, Adjusted R-squared:  0.1497
## F-statistic: 50.12 on 1 and 278 DF,  p-value: 1.183e-11
```

6 use HW performance (ranks) to predict midterm grades

6.1 use HW ranks to predict midterm 1

```
summary(lm(Midterm1 ~ HW0 + HW3 + HW4, data = grade_wide))
```

```
##
## Call:
## lm(formula = Midterm1 ~ HW0 + HW3 + HW4, data = grade_wide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.648  -9.647   3.118  11.900  37.413
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 63.651399   3.541026  17.975  < 2e-16 ***
## HW0          0.002365   0.019051   0.124  0.90130
## HW3          0.106965   0.013911   7.689 2.77e-13 ***
## HW4          0.044037   0.014163   3.109  0.00208 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.81 on 270 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.2756, Adjusted R-squared:  0.2676
## F-statistic: 34.25 on 3 and 270 DF,  p-value: < 2.2e-16
```

6.2 use HW ranks to predict midterm 2

```
##
## Call:
## lm(formula = Midterm2 ~ HW0 + HW3 + HW4 + HW5 + HW6 + HW7 + HW8,
##     data = grade_wide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.410  -7.787   2.004  10.685  44.601
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.278440   3.865987  13.523  < 2e-16 ***
## HW0         -0.006646   0.019172  -0.347 0.729134
## HW3          0.058831   0.016673   3.528 0.000492 ***
## HW4          0.019764   0.015011   1.317 0.189072
## HW5          0.023879   0.015903   1.501 0.134416
## HW6          0.073048   0.017453   4.186 3.87e-05 ***
## HW7          0.020857   0.015817   1.319 0.188419
## HW8          0.014946   0.018684   0.800 0.424468
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.82 on 266 degrees of freedom
##   (6 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.3578, Adjusted R-squared:  0.3409
## F-statistic: 21.17 on 7 and 266 DF,  p-value: < 2.2e-16
```