FirstProject

Rabin Paudel

11/15/2021

Problem 1 (30 pts)

```
library("stringr")
library("tidyverse")
```

You can extract the raw data for this table as follows.

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#install.packages("dplyr")
library(dplyr)
library(ggplot2)
library("pdftools") ## install.library("pdftools") if this library is missing
```

```
## Using poppler version 20.12.1
```

```
temp_file <- tempfile()
url <- paste0("https://www.pnas.org/content/suppl/2015/09/16/",
"1510159112.DCSupplemental/pnas.201510159SI.pdf")
download.file(url, temp_file)
txt <- pdf_text(temp_file)
file.remove(temp_file)
```

```
## [1] TRUE
```

```
## [1] TRUE
raw_data <- txt[2]
raw_data
```

```
## [1] "                        Table S1. Numbers of applications and awarded grants, along with success
```

First, use the function str_split from the stringr library to split the

```
raw_data_lines <- stringr::str_split(raw_data, "\n")[[1]]
head(raw_data_lines, 8)
```

raw_data string into lines. For example.

```
## [1] "                       Table S1. Numbers of applications and awarded grants, along with success
## [2] "                       female applicants, by scientific discipline"
## [3] "                                                  Applications, n                    Awards
## [4] ""
## [5] "                       Discipline              Total       Men      Women       Total      Men
## [6] ""
## [7] "                       Total                   2,823     1,635      1,188         467      290
## [8] "                       Chemical sciences         122        83         39          32       22
```

The numbers in the tables are recorded in the 7th through 15th

```
tab_data <- raw_data_lines[7:15]
tab_data
```

lines/elements of raw_data_lines.

```
## [1] "                       Total                   2,823     1,635      1,188         467      290
## [2] "                       Chemical sciences         122        83         39          32       22
## [3] "                       Physical sciences         174       135         39          35       26
## [4] "                       Physics                    76        67          9          20       18
## [5] "                       Humanities                396       230        166          65       33
## [6] "                       Technical sciences        251       189         62          43       30
## [7] "                       Interdisciplinary         183       105         78          29       12
## [8] "                       Earth/life sciences       282       156        126          56       38
## [9] "                       Social sciences           834       425        409         112       65
```

We can now try to use str_trim and str_split to split each line into

separate columns. We might want to take a careful look at the arguments

of str_split or try out the examples provided in the help page for

str_split.Then,After doing the above steps, you should now have something

resembling what we want. Now add the column names and remember to convert

the values in most of the columns into numbers. The function across in

dplyr might be useful here.

As always, refer to do this cheatsheet.

This is the required table.

```
tab_data1 <- str_trim(tab_data, side = c("both"))
tab_data2 <- str_split(tab_data1, "\\s{2,}", simplify = TRUE) %>%
  data.frame() %>%
  setNames(c('discipline', 'app_T', 'app_M', 'app_F', 'awards_T', 'awards_M',
             'awards_F', 'success_rates_T', 'success_rates_M',
             'success_rates_F')) %>% mutate_at(-1, parse_number) %>%
  as_tibble()
tab_data2
```

```
## # A tibble: 9 x 10
##   discipline        app_T app_M app_F awards_T awards_M awards_F success_rates_T
##   <chr>             <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>           <dbl>
## 1 Total              2823  1635  1188      467      290      177            16.5
## 2 Chemical sciences   122    83    39       32       22       10            26.2
## 3 Physical sciences   174   135    39       35       26        9            20.1
## 4 Physics              76    67     9       20       18        2            26.3
## 5 Humanities          396   230   166       65       33       32            16.4
## 6 Technical scienc~   251   189    62       43       30       13            17.1
## 7 Interdisciplinary   183   105    78       29       12       17            15.8
## 8 Earth/life scien~   282   156   126       56       38       18            19.9
## 9 Social sciences     834   425   409      112       65       47            13.4
## # ... with 2 more variables: success_rates_M <dbl>, success_rates_F <dbl>
```

Problem 2 (30pts)

The data had been studied in this article and is included as part the

```
library(SemiPar) ## install.packages("SemiPar") if this library is missing
data(milan.mort)
head(milan.mort)
```

SemiPar library in R.

```
##   day.num day.of.week holiday mean.temp rel.humid tot.mort resp.mort    SO2
## 1       1           2       1       5.6      30.0       45         2 267.33
## 2       2           3       0       4.1      26.0       32         5 374.98
## 3       3           4       0       4.6      29.7       37         0 276.25
## 4       4           5       0       2.9      32.7       33         1 440.50
```

```
## 5       5           6       0       2.2      71.3       36         1 354.25
## 6       6           7       0       0.7      80.7       45         6 334.50
##      TSP
## 1 109.56
## 2 152.68
## 3 162.16
## 4 197.52
## 5 234.59
## 6 167.34
```

```
milan.mort<- milan.mort %>%
  mutate(month = trunc((day.num/30 )%%12))
head(milan.mort)
```

Try to split day.num variable into month to pridect resp.mort

```
##   day.num day.of.week holiday mean.temp rel.humid tot.mort resp.mort    SO2
## 1       1           2       1       5.6      30.0       45         2 267.33
## 2       2           3       0       4.1      26.0       32         5 374.98
## 3       3           4       0       4.6      29.7       37         0 276.25
## 4       4           5       0       2.9      32.7       33         1 440.50
## 5       5           6       0       2.2      71.3       36         1 354.25
## 6       6           7       0       0.7      80.7       45         6 334.50
##      TSP month
## 1 109.56     0
## 2 152.68     0
## 3 162.16     0
## 4 197.52     0
## 5 234.59     0
## 6 167.34     0
```

```
train_idx <- sample(1:nrow(milan.mort), 0.8*nrow(milan.mort), replace = FALSE)
milan_mort_train <- milan.mort[train_idx,]
## Training data contains 80% of the observations
milan_mort_test <- milan.mort[-train_idx,]
## Testing data contains 20% of the observations
names(milan_mort_train)
```

Split the data into a random training and testing chunk.

```
##  [1] "day.num"     "day.of.week" "holiday"     "mean.temp"   "rel.humid"
##  [6] "tot.mort"    "resp.mort"   "SO2"         "TSP"         "month"
```

Try to change int variable into the factor and character variable like

```
milan_mort_train$holiday <- as.factor(milan_mort_train$holiday)
milan_mort_train$day.of.week <- as.factor(milan_mort_train$day.of.week)
milan_mort_train$month <- as.character(milan_mort_train$month)
str(milan_mort_train)
```

day.of.week and holiday.

```
## 'data.frame':    2921 obs. of  10 variables:
##  $ day.num    : int  3647 1270 161 1356 1098 3584 3506 618 1062 3487 ...
##  $ day.of.week: Factor w/ 7 levels "1","2","3","4",..: 1 4 1 6 7 1 7 3 6 2 ...
##  $ holiday    : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ mean.temp  : num  3.9 22 15.5 16.8 4.2 12.6 22.8 24 10.1 23.1 ...
##  $ rel.humid  : num  93 64.3 89 64.3 63 88 78 59.3 96 45.7 ...
##  $ tot.mort   : num  38 29 26 26 37 25 26 32 42 23 ...
##  $ resp.mort  : int  5 1 4 0 5 0 1 4 4 1 ...
##  $ SO2        : num  57.8 36.2 15.5 30 309 ...
##  $ TSP        : num  64.5 95 88.6 71 146.5 ...
##  $ month      : chr  "1" "6" "5" "9" ...
```

```
mod_naive <- lm(resp.mort ~ SO2 + TSP, milan_mort_train)
```

Fit model on training data

```
milan_mort_test_predict <- predict(mod_naive, milan_mort_test)
```

Predicted value on test data

```
mae_naive <- mean(abs(milan_mort_test_predict - milan_mort_test$resp.mort))
mae_naive
```

Mean absolute error of prediction

```
## [1] 1.255833
```

Your completed answer should have the following components.

You should consider two or three different models.

The first model is a very simple/simplistic model that serves as a naive

```
#pairs.panels(milan_mort_train)
```

baseline. Try to look relation between the variable.

We know resp.mort and tot.mort are highly correlated, and SO2 and TSP also correlated. The variable resp.mort is right skew.

First Model

```
mod0 <- lm(resp.mort ~ SO2 + TSP, milan_mort_train)
mod0
```

```
##
## Call:
## lm(formula = resp.mort ~ SO2 + TSP, data = milan_mort_train)
##
## Coefficients:
## (Intercept)          SO2          TSP
##    1.831797     0.005058    -0.001656
```

```
summary(mod0)
```

```
##
## Call:
## lm(formula = resp.mort ~ SO2 + TSP, data = milan_mort_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.957 -1.021 -0.076  1.018  8.203
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.8317965  0.0611892  29.937  < 2e-16 ***
## SO2          0.0050579  0.0003162  15.998  < 2e-16 ***
## TSP         -0.0016559  0.0005017  -3.301 0.000975 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.594 on 2918 degrees of freedom
## Multiple R-squared:  0.1029, Adjusted R-squared:  0.1023
## F-statistic: 167.3 on 2 and 2918 DF,  p-value: < 2.2e-16
```

```
AIC(mod0)
```

```
## [1] 11017.8
```

```
milan_mort_test_predict <- predict(mod0, milan_mort_test)
## Mean absolute error of prediction
mae_naive <- mean(abs(milan_mort_test_predict - milan_mort_test$resp.mort))
mae_naive
```

```
## [1] 1.255833
```

Second Model

The second model is a sufficiently complicated model (but should not have

```
mod1 <- lm(resp.mort ~SO2 + TSP + rel.humid + mean.temp, milan_mort_train)
mod1
```

say, more than 15 coeffi-cients).

```
##
## Call:
## lm(formula = resp.mort ~ SO2 + TSP + rel.humid + mean.temp, data = milan_mort_train)
##
## Coefficients:
## (Intercept)          SO2          TSP    rel.humid    mean.temp
##    2.813256     0.004033    -0.001463    -0.008626    -0.025519
```

```
AIC(mod1)
```

```
## [1] 10980.43
```

```
summary(mod1)
```

```
##
## Call:
## lm(formula = resp.mort ~ SO2 + TSP + rel.humid + mean.temp, data = milan_mort_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9090 -1.0847 -0.1398  0.9783  8.4903
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.8132557  0.1640481  17.149  < 2e-16 ***
## SO2          0.0040325  0.0003768  10.702  < 2e-16 ***
## TSP         -0.0014625  0.0005023  -2.912  0.00362 **
## rel.humid   -0.0086259  0.0017717  -4.869 1.18e-06 ***
## mean.temp   -0.0255194  0.0049537  -5.152 2.75e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.583 on 2916 degrees of freedom
## Multiple R-squared:  0.1155, Adjusted R-squared:  0.1143
## F-statistic:  95.2 on 4 and 2916 DF,  p-value: < 2.2e-16
```

```
AIC(mod1)
```

```
## [1] 10980.43
```

Third Model

```
#Try to remove non significance variable
mod2 <- lm(resp.mort ~.-TSP, milan_mort_train)
mod2
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP, data = milan_mort_train)
##
## Coefficients:
##  (Intercept)       day.num  day.of.week2  day.of.week3  day.of.week4
##   -3.801e-01    -9.746e-05    -1.869e-01    -1.646e-01    -2.331e-01
## day.of.week5  day.of.week6  day.of.week7      holiday1     mean.temp
##   -1.867e-01    -1.177e-01    -1.435e-01    -1.063e-02     2.051e-02
##    rel.humid      tot.mort           SO2        month1       month10
##   -2.874e-03     8.087e-02     1.639e-03     4.740e-01    -2.379e-01
##      month11        month2        month3        month4        month5
##   -1.597e-01     7.049e-01     4.886e-01     7.022e-03    -1.190e-01
##       month6        month7        month8        month9
##   -2.896e-01    -2.302e-01    -7.079e-02    -3.280e-01
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP, data = milan_mort_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0545 -1.0229 -0.1504  0.8484  5.9855
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.801e-01  2.724e-01  -1.396 0.162905
## day.num      -9.746e-05  3.028e-05  -3.219 0.001301 **
## day.of.week2 -1.869e-01  1.001e-01  -1.866 0.062104 .
## day.of.week3 -1.646e-01  1.004e-01  -1.640 0.101112
## day.of.week4 -2.331e-01  1.001e-01  -2.329 0.019916 *
## day.of.week5 -1.867e-01  9.988e-02  -1.869 0.061732 .
## day.of.week6 -1.177e-01  9.920e-02  -1.187 0.235405
## day.of.week7 -1.435e-01  1.006e-01  -1.426 0.153918
```

```
## holiday1     -1.063e-02  1.607e-01  -0.066 0.947275
## mean.temp     2.051e-02  8.260e-03   2.483 0.013079 *
## rel.humid    -2.874e-03  1.704e-03  -1.687 0.091707 .
## tot.mort      8.087e-02  4.244e-03  19.056  < 2e-16 ***
## SO2           1.639e-03  3.810e-04   4.303 1.74e-05 ***
## month1        4.740e-01  1.260e-01   3.761 0.000173 ***
## month10      -2.379e-01  1.544e-01  -1.541 0.123518
## month11      -1.597e-01  1.333e-01  -1.198 0.230976
## month2        7.049e-01  1.320e-01   5.339 1.01e-07 ***
## month3        4.886e-01  1.449e-01   3.372 0.000757 ***
## month4        7.022e-03  1.595e-01   0.044 0.964898
## month5       -1.190e-01  1.795e-01  -0.663 0.507543
## month6       -2.896e-01  1.936e-01  -1.496 0.134770
## month7       -2.302e-01  2.113e-01  -1.089 0.276088
## month8       -7.079e-02  2.072e-01  -0.342 0.732579
## month9       -3.280e-01  1.851e-01  -1.772 0.076479 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.451 on 2897 degrees of freedom
## Multiple R-squared:  0.2615, Adjusted R-squared:  0.2556
## F-statistic: 44.59 on 23 and 2897 DF,  p-value: < 2.2e-16
```

```
library(forcats)
# Try to remove non significance variable
milan_mort_train %>% select(day.of.week) %>% pull() %>% table
```

```
## .
##   1   2   3   4   5   6   7
## 431 412 410 413 416 430 409
```

```
milan_mort_train %>% select(month) %>% pull() %>% table
```

```
## .
##   0   1  10  11   2   3   4   5   6   7   8   9
## 270 266 228 246 241 233 237 235 250 243 236 236
```

```
milan_mort_train1 <- milan_mort_train%>% mutate(day.of.week =
                                      fct_collapse(day.of.week,
milan_mort_train2 <- milan_mort_train1 %>% mutate(month =
                        fct_collapse(month,
                            "0" = c("4","5", "6", "7", "8", "9", "10", "11")))
head(milan_mort_train2)
```

```
##      day.num day.of.week holiday mean.temp rel.humid tot.mort resp.mort    SO2
## 3647    3647           1       1       3.9      93.0       38         5  57.75
## 1270    1270           4       0      22.0      64.3       29         1  36.25
## 161      161           1       0      15.5      89.0       26         4  15.54
## 1356    1356           3       0      16.8      64.3       26         0  30.00
## 1098    1098           3       0       4.2      63.0       37         5 309.00
## 3584    3584           1       0      12.6      88.0       25         0  78.07
##         TSP month
```

```
## 3647  64.53     1
## 1270  94.99     0
## 161   88.60     0
## 1356  71.00     0
## 1098 146.50     0
## 3584 150.50     0
```

```
mod2 <- lm(resp.mort ~.-TSP-holiday-mean.temp-tot.mort - rel.humid,
          milan_mort_train2)
mod2
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP - holiday - mean.temp - tot.mort -
##     rel.humid, data = milan_mort_train2)
##
## Coefficients:
##  (Intercept)       day.num  day.of.week2  day.of.week3  day.of.week4
##     2.238397     -0.000219     -0.169887     -0.195586     -0.251244
##          SO2        month1        month2        month3
##     0.002638      0.778652      1.011265      0.801614
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP - holiday - mean.temp - tot.mort -
##     rel.humid, data = milan_mort_train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6123 -1.0809 -0.1853  0.9005  8.1907
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.238e+00  1.024e-01  21.851  < 2e-16 ***
## day.num      -2.190e-04  2.935e-05  -7.463 1.11e-13 ***
## day.of.week2 -1.699e-01  1.066e-01  -1.594   0.1111
## day.of.week3 -1.956e-01  8.365e-02  -2.338   0.0194 *
## day.of.week4 -2.512e-01  1.065e-01  -2.358   0.0184 *
## SO2           2.638e-03  2.849e-04   9.259  < 2e-16 ***
## month1        7.787e-01  1.117e-01   6.969 3.92e-12 ***
## month2        1.011e+00  1.083e-01   9.337  < 2e-16 ***
## month3        8.016e-01  1.068e-01   7.508 7.93e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.547 on 2912 degrees of freedom
## Multiple R-squared:  0.1569, Adjusted R-squared:  0.1546
## F-statistic: 67.73 on 8 and 2912 DF,  p-value: < 2.2e-16
```

```
AIC(mod2)
```

```
## [1] 10848.51
```

```
#Try to remove non significance variable
mod2 <- lm(resp.mort ~.-TSP, milan_mort_train)
mod2
```

After removing non significant value from the model.

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP, data = milan_mort_train)
##
## Coefficients:
##  (Intercept)       day.num  day.of.week2  day.of.week3  day.of.week4
##   -3.801e-01    -9.746e-05    -1.869e-01    -1.646e-01    -2.331e-01
## day.of.week5  day.of.week6  day.of.week7      holiday1     mean.temp
##   -1.867e-01    -1.177e-01    -1.435e-01    -1.063e-02     2.051e-02
##    rel.humid      tot.mort           SO2        month1       month10
##   -2.874e-03     8.087e-02     1.639e-03     4.740e-01    -2.379e-01
##      month11        month2        month3        month4        month5
##   -1.597e-01     7.049e-01     4.886e-01     7.022e-03    -1.190e-01
##       month6        month7        month8        month9
##   -2.896e-01    -2.302e-01    -7.079e-02    -3.280e-01
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP, data = milan_mort_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0545 -1.0229 -0.1504  0.8484  5.9855
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.801e-01  2.724e-01  -1.396 0.162905
## day.num      -9.746e-05  3.028e-05  -3.219 0.001301 **
## day.of.week2 -1.869e-01  1.001e-01  -1.866 0.062104 .
## day.of.week3 -1.646e-01  1.004e-01  -1.640 0.101112
## day.of.week4 -2.331e-01  1.001e-01  -2.329 0.019916 *
## day.of.week5 -1.867e-01  9.988e-02  -1.869 0.061732 .
## day.of.week6 -1.177e-01  9.920e-02  -1.187 0.235405
## day.of.week7 -1.435e-01  1.006e-01  -1.426 0.153918
## holiday1     -1.063e-02  1.607e-01  -0.066 0.947275
## mean.temp     2.051e-02  8.260e-03   2.483 0.013079 *
## rel.humid    -2.874e-03  1.704e-03  -1.687 0.091707 .
```

```
## tot.mort      8.087e-02  4.244e-03  19.056  < 2e-16 ***
## SO2           1.639e-03  3.810e-04   4.303 1.74e-05 ***
## month1        4.740e-01  1.260e-01   3.761 0.000173 ***
## month10      -2.379e-01  1.544e-01  -1.541 0.123518
## month11      -1.597e-01  1.333e-01  -1.198 0.230976
## month2        7.049e-01  1.320e-01   5.339 1.01e-07 ***
## month3        4.886e-01  1.449e-01   3.372 0.000757 ***
## month4        7.022e-03  1.595e-01   0.044 0.964898
## month5       -1.190e-01  1.795e-01  -0.663 0.507543
## month6       -2.896e-01  1.936e-01  -1.496 0.134770
## month7       -2.302e-01  2.113e-01  -1.089 0.276088
## month8       -7.079e-02  2.072e-01  -0.342 0.732579
## month9       -3.280e-01  1.851e-01  -1.772 0.076479 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.451 on 2897 degrees of freedom
## Multiple R-squared:  0.2615, Adjusted R-squared:  0.2556
## F-statistic: 44.59 on 23 and 2897 DF,  p-value: < 2.2e-16
```

```
library(forcats)
# Try to remove non significance variable
#milan_mort_train %>% select(day.of.week) %>% pull() %>% table
milan_mort_train %>% select(month) %>% pull() %>% table
```

```
## .
##   0   1  10  11   2   3   4   5   6   7   8   9
## 270 266 228 246 241 233 237 235 250 243 236 236
```

```
#milan_mort_train1 <- milan_mort_train%>% mutate(day.of.week =
#                                      fct_collapse(day.of.week,
milan_mort_train2 <- milan_mort_train %>% mutate(month =
                        fct_collapse(month,
                            "0" = c("4","5", "6", "7", "8", "9", "10", "11")))
head(milan_mort_train2)
```

```
##      day.num day.of.week holiday mean.temp rel.humid tot.mort resp.mort    SO2
## 3647    3647           1       1       3.9      93.0       38         5  57.75
## 1270    1270           4       0      22.0      64.3       29         1  36.25
## 161      161           1       0      15.5      89.0       26         4  15.54
## 1356    1356           6       0      16.8      64.3       26         0  30.00
## 1098    1098           7       0       4.2      63.0       37         5 309.00
## 3584    3584           1       0      12.6      88.0       25         0  78.07
##         TSP month
## 3647  64.53     1
## 1270  94.99     0
## 161   88.60     0
## 1356  71.00     0
## 1098 146.50     0
## 3584 150.50     0
```

```
mod2 <- lm(resp.mort ~.-TSP-holiday-mean.temp-tot.mort -day.of.week- rel.humid,
           milan_mort_train2)
mod2
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP - holiday - mean.temp - tot.mort -
##     day.of.week - rel.humid, data = milan_mort_train2)
##
## Coefficients:
## (Intercept)      day.num          SO2       month1       month2       month3
##   2.0680195   -0.0002195    0.0026491    0.7708901    1.0103854    0.7973267
```

```
summary(mod2)
```

```
##
## Call:
## lm(formula = resp.mort ~ . - TSP - holiday - mean.temp - tot.mort -
##     day.of.week - rel.humid, data = milan_mort_train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6335 -1.0724 -0.1830  0.9023  8.1658
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.068e+00  7.585e-02  27.264  < 2e-16 ***
## day.num     -2.195e-04  2.937e-05  -7.473 1.03e-13 ***
## SO2          2.649e-03  2.849e-04   9.298  < 2e-16 ***
## month1       7.709e-01  1.117e-01   6.898 6.42e-12 ***
## month2       1.010e+00  1.084e-01   9.324  < 2e-16 ***
## month3       7.973e-01  1.068e-01   7.464 1.10e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.548 on 2915 degrees of freedom
## Multiple R-squared:  0.1549, Adjusted R-squared:  0.1535
## F-statistic: 106.9 on 5 and 2915 DF,  p-value: < 2.2e-16
```

```
AIC(mod2)
```

```
## [1] 10849.31
```

```
## Average number of deaths per day using the training data
estimate <- mean(milan_mort_train2$resp.mort)
estimate
```

```
## [1] 2.187607
```

```
value <- mean(abs(estimate - milan_mort_test$resp.mort))
value
```

```
## [1] 1.331338
```

(2) You should provide some brief discussion on.

(a) why you include these predictor variables in your model

ANS: My best model is model 3 and my predicted variable day.num and three month period Jan, Feb, march, and SO2 gas these variable are highely significant. The model three also AIC and residual standard error are small than other two model, so model three is best fit. Overall F test and p-value both highly significant. My conclusion is that in highly SO2 gas in air in the cold season are more problematic more people are died for respiratory disease.

(b) how satisfied you are with the accuracy of your model and.

ANS: I think my model is good enough through the given data set.

I have no idea why R^2 and AdjR^2 are so small.

(c) if appropriate) are there any issues with your model and what will you try/do if you have more time (for example, there could be seasonal trend, the response are integers not real numbers, . . .)

ANS: Yes, I converted some variable into the factor as well as char variable.

I convert day number as a month variable to figure out weather is important or not,but I found weather is important cold season more problem of respiratory related disease.I try to research day.of.week variable,

but these variable not significant.

Problem 3(15pts)

```
df <- readr::read_table("http://stat.rutgers.edu/home/mxie/stat586/homework/hw2_junka_data.txt")
```

The data is available here and can be read into R as follows

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Dens = col_double(),
##   Hard = col_double()
## )
```

```
df
```

```
## # A tibble: 36 x 2
##     Dens  Hard
##    <dbl> <dbl>
##  1  24.7   484
##  2  24.8   427
##  3  27.3   413
##  4  28.4   517
##  5  28.4   549
##  6  29     648
##  7  30.3   587
##  8  32.7   704
##  9  35.6   979
## 10  38.5   914
## # ... with 26 more rows
```

```
basis.design <- function(x,knots){
Xmat <- cbind(rep(1, length(x)), x)
for(i in 1:length(knots)){
Xmat <- cbind(Xmat, ifelse(x < knots[i], 0, x - knots[i]))
}
return( Xmat)
}
psr <- function(y,x,knots,lambda){
X <- basis.design(x, knots)
D <- diag(c(0,0, rep(1,length(knots))))
S <- X %*% solve(t(X) %*% X + lambda*D) %*% t(X)
yhat <- S %*% y
df_fit <- sum(diag(S)) ## degrees of freedom for the fit
n <- nrow(X)
gcv <- sum((y - yhat)^2)/(1 - df_fit/n)^2 ## Generalized cross-validation score
return(list(yhat = yhat, gcv = gcv)) ## Return the fitted value and the gcv score
}
```

For this problem, see the lecture slides on penalized splines. In particular the following two functions

The (solid) regression line and (dashed) curve in the above plot correspond

to a simple least square regression line log(hardness) 0 + 1density #### and a nonparametric regression line, respectively. The nonparametric #### regression line is fitted via penalized spline regression of the form

```
knots <- seq(from = 0, to = 70, by = 10) # use the function.
val <- psr(log(df$Hard), df$Dens, knots,1) # create val.
val
```

```
## $yhat
##           [,1]
##  [1,] 6.069431
##  [2,] 6.075694
##  [3,] 6.232288
##  [4,] 6.301189
##  [5,] 6.301189
##  [6,] 6.338772
##  [7,] 6.419001
##  [8,] 6.559741
##  [9,] 6.729801
## [10,] 6.899861
## [11,] 6.917453
## [12,] 6.946774
## [13,] 6.952638
## [14,] 6.981959
## [15,] 7.001591
## [16,] 7.015358
## [17,] 7.019947
## [18,] 7.019947
## [19,] 7.120909
## [20,] 7.253995
## [21,] 7.304475
## [22,] 7.364135
## [23,] 7.480416
## [24,] 7.480416
## [25,] 7.523073
## [26,] 7.581446
## [27,] 7.592672
## [28,] 7.610633
## [29,] 7.617368
## [30,] 7.653290
## [31,] 7.666761
## [32,] 7.895997
## [33,] 7.948437
## [34,] 8.000878
## [35,] 8.012115
## [36,] 8.012115
##
```

```
## $gcv
## [1] 0.4284626
```

```
newData <- data.frame(df,val) # Create new data frame.
names(newData)
```

```
## [1] "Dens" "Hard" "yhat" "gcv"
```

```
newData
```

```
##    Dens Hard     yhat       gcv
## 1  24.7  484 6.069431 0.4284626
## 2  24.8  427 6.075694 0.4284626
## 3  27.3  413 6.232288 0.4284626
## 4  28.4  517 6.301189 0.4284626
## 5  28.4  549 6.301189 0.4284626
## 6  29.0  648 6.338772 0.4284626
## 7  30.3  587 6.419001 0.4284626
## 8  32.7  704 6.559741 0.4284626
## 9  35.6  979 6.729801 0.4284626
## 10 38.5  914 6.899861 0.4284626
## 11 38.8 1070 6.917453 0.4284626
## 12 39.3 1020 6.946774 0.4284626
## 13 39.4 1210 6.952638 0.4284626
## 14 39.9  989 6.981959 0.4284626
## 15 40.3 1160 7.001591 0.4284626
## 16 40.6 1010 7.015358 0.4284626
## 17 40.7 1100 7.019947 0.4284626
## 18 40.7 1130 7.019947 0.4284626
## 19 42.9 1270 7.120909 0.4284626
## 20 45.8 1180 7.253995 0.4284626
## 21 46.9 1400 7.304475 0.4284626
## 22 48.2 1760 7.364135 0.4284626
## 23 51.5 1710 7.480416 0.4284626
## 24 51.5 2010 7.480416 0.4284626
## 25 53.4 1880 7.523073 0.4284626
## 26 56.0 1980 7.581446 0.4284626
## 27 56.5 1820 7.592672 0.4284626
## 28 57.3 2020 7.610633 0.4284626
## 29 57.6 1980 7.617368 0.4284626
## 30 59.2 2310 7.653290 0.4284626
## 31 59.8 1940 7.666761 0.4284626
## 32 66.0 3260 7.895997 0.4284626
## 33 67.4 2700 7.948437 0.4284626
## 34 68.8 2890 8.000878 0.4284626
## 35 69.1 2740 8.012115 0.4284626
## 36 69.1 3140 8.012115 0.4284626
```

```
# Try to create the model.
mu_x <- mean(newData$Dens)
mu_y <- mean(log(newData$Hard))
s_x <- sd(newData$Dens)
s_x
```

```
## [1] 13.58009
```

```
s_y <- sd(log(newData$Hard))
r <- cor(newData$Dens,log(newData$Hard))
r
```

```
## [1] 0.9737636
```

```
fit <- lm(log(newData$Hard) ~ Dens + yhat , data = newData) # Fit the line.
summary(fit)
```

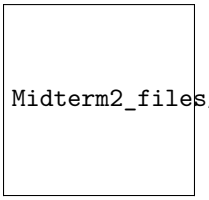
```
##
## Call:
## lm(formula = log(newData$Hard) ~ Dens + yhat, data = newData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20847 -0.05478 -0.01002  0.05517  0.19376
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0152980  0.8732547  -0.018    0.986
## Dens        -0.0001238  0.0071507  -0.017    0.986
## yhat         1.0029371  0.1673052   5.995  9.8e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09531 on 33 degrees of freedom
## Multiple R-squared:  0.9752, Adjusted R-squared:  0.9737
## F-statistic: 649.1 on 2 and 33 DF,  p-value: < 2.2e-16
```

```
df <- newData %>% mutate(rp = predict(fit))
df
```

```
##    Dens Hard     yhat       gcv       rp
## 1  24.7  484 6.069431 0.4284626 6.068901
## 2  24.8  427 6.075694 0.4284626 6.075171
## 3  27.3  413 6.232288 0.4284626 6.231915
## 4  28.4  517 6.301189 0.4284626 6.300882
## 5  28.4  549 6.301189 0.4284626 6.300882
## 6  29.0  648 6.338772 0.4284626 6.338501
## 7  30.3  587 6.419001 0.4284626 6.418806
## 8  32.7  704 6.559741 0.4284626 6.559661
## 9  35.6  979 6.729801 0.4284626 6.729862
## 10 38.5  914 6.899861 0.4284626 6.900062
## 11 38.8 1070 6.917453 0.4284626 6.917669
## 12 39.3 1020 6.946774 0.4284626 6.947014
## 13 39.4 1210 6.952638 0.4284626 6.952883
## 14 39.9  989 6.981959 0.4284626 6.982228
## 15 40.3 1160 7.001591 0.4284626 7.001868
## 16 40.6 1010 7.015358 0.4284626 7.015639
## 17 40.7 1100 7.019947 0.4284626 7.020229
```

```
## 18 40.7 1130 7.019947 0.4284626 7.020229
## 19 42.9 1270 7.120909 0.4284626 7.121215
## 20 45.8 1180 7.253995 0.4284626 7.254332
## 21 46.9 1400 7.304475 0.4284626 7.304825
## 22 48.2 1760 7.364135 0.4284626 7.364499
## 23 51.5 1710 7.480416 0.4284626 7.480713
## 24 51.5 2010 7.480416 0.4284626 7.480713
## 25 53.4 1880 7.523073 0.4284626 7.523261
## 26 56.0 1980 7.581446 0.4284626 7.581483
## 27 56.5 1820 7.592672 0.4284626 7.592680
## 28 57.3 2020 7.610633 0.4284626 7.610595
## 29 57.6 1980 7.617368 0.4284626 7.617313
## 30 59.2 2310 7.653290 0.4284626 7.653142
## 31 59.8 1940 7.666761 0.4284626 7.666578
## 32 66.0 3260 7.895997 0.4284626 7.895719
## 33 67.4 2700 7.948437 0.4284626 7.948141
## 34 68.8 2890 8.000878 0.4284626 8.000562
## 35 69.1 2740 8.012115 0.4284626 8.011795
## 36 69.1 3140 8.012115 0.4284626 8.011795
```

```
#Using the ggplot and plot the data.
#This is required chart.
ggplot(df, aes(Dens, log(Hard))) + geom_point() +
  geom_line(aes(Dens, rp), color = 'blue', linetype = 'dashed') +
  geom_abline(slope = r * s_y/s_x, intercept = mu_y - r * s_y/s_x * mu_x,color = 'blue')
```



Midterm2_files/figure-latex/unnamed-chunk-20-1.pdf

THE END

THANK YOU!