Biostat_Methods_Final_Project

Junyuan Zheng (jz3036)

2018-12-07

Import packages:

==

DATA Preprocessing

Import original dataset:

build 'region' variable from 'geography'. This is gonna be the only categorical variable in the dataset.

```
data_pre = data %>%
separate(., geography, into = c('county', 'state'), sep = ', ') %>%
  mutate(., region = replace(state, state == 'Connecticut', 'Northeast'),
            region = replace(region, region == 'Maine', 'Northeast'),
            region = replace(region, region == 'Massachusetts', 'Northeast'),
            region = replace(region, region == 'New Hampshire', 'Northeast'),
            region = replace(region, region == 'Rhode Island', 'Northeast'),
            region = replace(region, region == 'Vermont', 'Northeast'),
            region = replace(region, region == 'New Jersey', 'Northeast'),
            region = replace(region, region == 'New York', 'Northeast'),
            region = replace(region, region == 'Pennsylvania', 'Northeast'),
            region = replace(region, region == 'Indiana', 'Midwest'),
            region = replace(region, region == 'Michigan', 'Midwest'),
            region = replace(region, region == 'Illinois', 'Midwest'),
            region = replace(region, region == 'Ohio', 'Midwest'),
            region = replace(region, region == 'Wisconsin', 'Midwest'),
            region = replace(region, region == 'Iowa', 'Midwest'),
            region = replace(region, region == 'Kansas', 'Midwest'),
            region = replace(region, region == 'Minnesota', 'Midwest'),
            region = replace(region, region == 'Missouri', 'Midwest'),
            region = replace(region, region == 'Nebraska', 'Midwest'),
            region = replace(region, region == 'North Dakota', 'Midwest'),
            region = replace(region, region == 'South Dakota', 'Midwest'),
            region = replace(region, region == 'Delaware', 'South'),
            region = replace(region, region == 'Florida', 'South'),
            region = replace(region, region == 'Georgia', 'South'),
            region = replace(region, region == 'Maryland', 'South'),
            region = replace(region, region == 'North Carolina', 'South'),
            region = replace(region, region == 'South Carolina', 'South'),
            region = replace(region, region == 'Virginia', 'South'),
            region = replace(region, region == 'District of Columbia', 'South'),
            region = replace(region, region == 'West Virginia', 'South'),
            region = replace(region, region == 'Alabama', 'South'),
            region = replace(region, region == 'Kentucky', 'South'),
            region = replace(region, region == 'Mississippi', 'South'),
            region = replace(region, region == 'Tennessee', 'South'),
            region = replace(region, region == 'Arkansas', 'South'),
            region = replace(region, region == 'Louisiana', 'South'),
            region = replace(region, region == 'Oklahoma', 'South'),
            region = replace(region, region == 'Texas', 'South'),
            region = replace(region, region == 'Arizona', 'West'),
            region = replace(region, region == 'Colorado', 'West'),
            region = replace(region, region == 'Idaho', 'West'),
            region = replace(region, region == 'Montana', 'West'),
            region = replace(region, region == 'Nevada', 'West'),
            region = replace(region, region == 'New Mexico', 'West'),
            region = replace(region, region == 'Utah', 'West'),
            region = replace(region, region == 'Wyoming', 'West'),
            region = replace(region, region == 'Alaska', 'West'),
            region = replace(region, region == 'California', 'West'),
            region = replace(region, region == 'Hawaii', 'West'),
            region = replace(region, region == 'Oregon', 'West'),
            region = replace(region, region == 'Washington', 'West'))
```

There are 3 variables that has missing data in the original dataset: ‘pct_some_col18_24’, ‘pct_employed16_over’, ‘pct_private_coverage_alone’. Since ‘pct_no_hs18_24’, ‘pct_some_col18_24’, ‘pct_hs18_24’, and ‘pct_bach_deg18_24’ together equals 100, here I re-calculate the ‘pct_some_col18_24’ variable so that it would not contain missing data anymore.

```
data_pre =
  mutate(data_pre, pct_some_col18_24 = 100 - (pct_no_hs18_24 + pct_hs18_24 + pct_bach_deg18_24))
```

==

Variable Pre-Selection (before putting variables into the model)

Things being considered under this section:
(1) Checking collinearity for the continuous variables (using 0.7 as rule of thumb).
(2) complete variable over variable with missing data.
(3) Practical importance.
(4) Real life situation.
(5) Not to eliminate too many variables for later model building (LASSO, Automatic selection).

```
data_pre %>%
  select(., -'binned_inc', -'county', -'state', -'region') %>%
  cor(., use='complete.obs')
```

decision reasoning for this part:

Decision 1: not to use ‘avg_deaths_per_year’ variable.
Reasoning: ‘avg_deaths_per_year’, ‘avg_ann_count’, and ‘pop_est2015’ appear to be correlated. First of all, it’s not hard for local government to have access to all the three variables, so it’s probably better to keep them all for now. However, since we are predicting the Death Rate, having both ‘avg_deaths_per_year’ and ‘pop_est2015’ available seems a little bit weird because we can probably just calculate the true Death Rate. Since population is a more common variable to have access to, I would not be using ‘avg_deaths_per_year’ during nest step.
Decision 2: keeping the ‘median_age_male’, ‘median_age_female’ pair, as well as the ‘pct_white’, ‘pct_black’ pair even they are correlated.

Reasoning: For practical importance, all of the four variables are common and what people are care about.
Decision 3: not to use 'binned_inc'. Reasoning: 'binned_inc' is the median income binned by decile, there is 10 distinct range for this variable, which means we can potentially make it to be a categorical data. However, this categorial variable might be complicated to interpret. Instead, I choose to use the continuous variable 'med_income' who is correlated to 'binned_inc'.
Decision 4: not to used 'pct_married_households' Reasoning: 'pct_married_households' and 'percent_married' are correlated, while 'percent_married' is more accessible.
Decision 5: keep a 'correlation network' of variables Reasoning: When looking at which variables are correlated, there is a group of variables, roughly including 'poverty_percent', 'med_income', the education_related variables,the employment_related variables, and the health coverage_related variables, that are somewhat correlated. I don't want to eliminate any of them just yet because, first of all, it's hard to decide which severals to keep, and secondly, I can live them for latter model building using LASSO or automatic selection method.
Decision 6: not to use 'pct_employed16_over' and 'pct_private_coverage_alone' with missing data. Reasoning: I didn't come up with some way to makeup of those missing data. If using this two variable, a lot of observations would not be able to fit in the model, therefore wasting some of the information. Also, since the two variables are somewhat correlated to other variables, I'm not too worried about missing those infomation.

Conclusion:

Variables not gonna be used before model selection:
'avg_deaths_per_year', 'binned_inc', 'pct_married_households', 'pct_employed16_over', 'pct_private_coverage_alone'.

Dataset after variable pre-selection:

```
data_vrb_pre_selec = data_pre %>%
  select(., -'avg_deaths_per_year', -'binned_inc', -'pct_married_households',
          -'pct_employed16_over', -'pct_private_coverage_alone', -'county', -'state')
```

===

Model Selection

LASSO

```
set.seed(1)
data_vrb_pre_selec_df = data.frame(data_vrb_pre_selec)

Y = data_vrb_pre_selec_df[,2]
X = model.matrix(target_death_rate ~ ., data=data_vrb_pre_selec_df) # 'region' into dummy variable.

train = sample(1:nrow(X),nrow(X)*0.8)

cv.out = cv.glmnet(X[train,],Y[train])
#plot(cv.out)
best.lambda = cv.out$lambda.min

lasso2 = glmnet(X, Y, alpha=1, lambda=best.lambda)
coef(lasso2)
```

```
## 32 x 1 sparse Matrix of class "dgCMatrix"
##                                      s0
## (Intercept)                1.111315e+02
## (Intercept)                .           
## avg_ann_count             -4.098831e-04
## incidence_rate             1.923068e-01
## med_income                 .           
## pop_est2015                .           
## poverty_percent            2.427210e-01
## study_per_cap              .           
## median_age                 .           
## median_age_male           -2.333928e-02
## median_age_female         -1.691406e-01
## avg_household_size        -5.434806e-01
## percent_married           -1.918078e-01
## pct_no_hs18_24            -7.493338e-02
## pct_hs18_24                2.410050e-01
## pct_some_col18_24          .           
## pct_bach_deg18_24         -3.496872e-02
## pct_hs25_over              3.791747e-01
## pct_bach_deg25_over       -8.259126e-01
## pct_unemployed16_over      5.453695e-01
## pct_private_coverage      -2.929637e-01
## pct_emp_priv_coverage      1.725788e-01
## pct_public_coverage        7.152181e-02
## pct_public_coverage_alone  3.951541e-01
## pct_white                 -1.129789e-01
## pct_black                 -1.635845e-01
## pct_asian                  1.036524e-01
## pct_other_race            -7.066427e-01
## birth_rate                -5.341927e-01
## regionNortheast           -8.225357e+00
## regionSouth                4.769389e+00
## regionWest                -9.919057e+00
```

Using LASSO, the coefficient of 'med_income', 'pop_est2015', 'study_per_cap', 'median_age' and 'pct_some_col18_24' became 0. Those variables can potentially be eliminated from our model.

Automatic Procedures

Forward Elimination

```
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
```

```
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
fit1 = lm(target_death_rate ~ pct_emp_priv_coverage+pct_private_coverage+pct_bach_deg18_24+pct_no_hs18_
tidy(fit1)
```

Forward Elimination result: 15 variables
lm(formula = target_death_rate ~ pct_bach_deg25_over + incidence_rate + poverty_percent + pct_other_race + region + pct_public_coverage_alone + pct_hs25_over + pct_unemployed16_over + median_age_female + birth_rate + pct_some_col18_24 + pct_no_hs18_24 + pct_bach_deg18_24 + pct_private_coverage + pct_emp_priv_coverage, data = data_vrb_pre_selec)

Backward Elimination

```
mult.fit_BW = lm(target_death_rate ~ ., data=data_vrb_pre_selec)
summary(mult.fit_BW)

step1 = update(mult.fit_BW, . ~ . -pct_bach_deg18_24)
summary(step1)

step2 = update(step1, . ~ . -study_per_cap)
summary(step2)

step3 = update(step2, . ~ . -med_income)
summary(step3)

step4 = update(step3, . ~ . -median_age)
summary(step4)

step5 = update(step4, . ~ . -pct_public_coverage_alone)
summary(step5)

step6 = update(step5, . ~ . -pct_asian)
summary(step6)

step7 = update(step6, . ~ . -pct_some_col18_24)
summary(step7)

step8 = update(step7, . ~ . -median_age_male)
summary(step8)

step9 = update(step8, . ~ . -avg_household_size)
summary(step9)
```

```
step10 = update(step9, . ~ . -pop_est2015)
summary(step10)

step11 = update(step10, . ~ . -avg_ann_count)
summary(step11)

step12 = update(step11, . ~ . -pct_no_hs18_24)
summary(step12)
```

Using backward selection, ‘pct_bach_deg18_24’, ‘study_per_cap’, ‘med_income’, ‘median_age’, ‘pct_public_coverage_alone’, ‘pct_asian’, ‘pct_some_col18_24’, ‘median_age_male’, ‘avg_household_size’, ‘pop_est2015’, ‘avg_ann_count’, and ‘pct_no_hs18_24’ were eliminated from the model.
lm(formula = target_death_rate ~ incidence_rate + poverty_percent + median_age_female + percent_married + pct_hs18_24 + pct_hs25_over + pct_bach_deg25_over + pct_unemployed16_over + pct_private_coverage + pct_emp_priv_coverage + pct_public_coverage + pct_white + pct_black + pct_other_race + birth_rate + region, data = data_vrb_pre_selec)

Stepwise Regression

```
mult.fit_SW = lm(target_death_rate ~ ., data=data_vrb_pre_selec)
step(mult.fit_SW, direction='backward')
```

Stepwise Result: 19 variables.
lm(formula = target_death_rate ~ avg_ann_count + incidence_rate + pop_est2015 + poverty_percent + median_age_female + percent_married + pct_no_hs18_24 + pct_hs18_24 + pct_hs25_over + pct_bach_deg25_over + pct_unemployed16_over + pct_private_coverage + pct_emp_priv_coverage + pct_public_coverage + pct_white + pct_black + pct_other_race + birth_rate + region, data = data_vrb_pre_selec) %>% summary()

criterion-based procedures

Cp

```
data_frame = data.frame(data_vrb_pre_selec)
leaps(x = data_frame[,c('avg_ann_count','incidence_rate','med_income','pop_est2015','poverty_percent','
```

Including ‘pct_bach_deg18_24’ doesn’t work for the function.

Suggested: 16 variables. ‘avg_ann_count’,‘incidence_rate’,‘pop_est2015’,‘poverty_percent’, ‘median_age_male’, ‘pct_no_hs18_24’,‘pct_hs18_24’, ‘pct_hs25_over’,‘pct_bach_deg25_over’,‘pct_unemployed16_over’,‘pct_

adjr2:

```
leaps(x = data_frame[,c('avg_ann_count','incidence_rate','med_income','pop_est2015','poverty_percent','
```

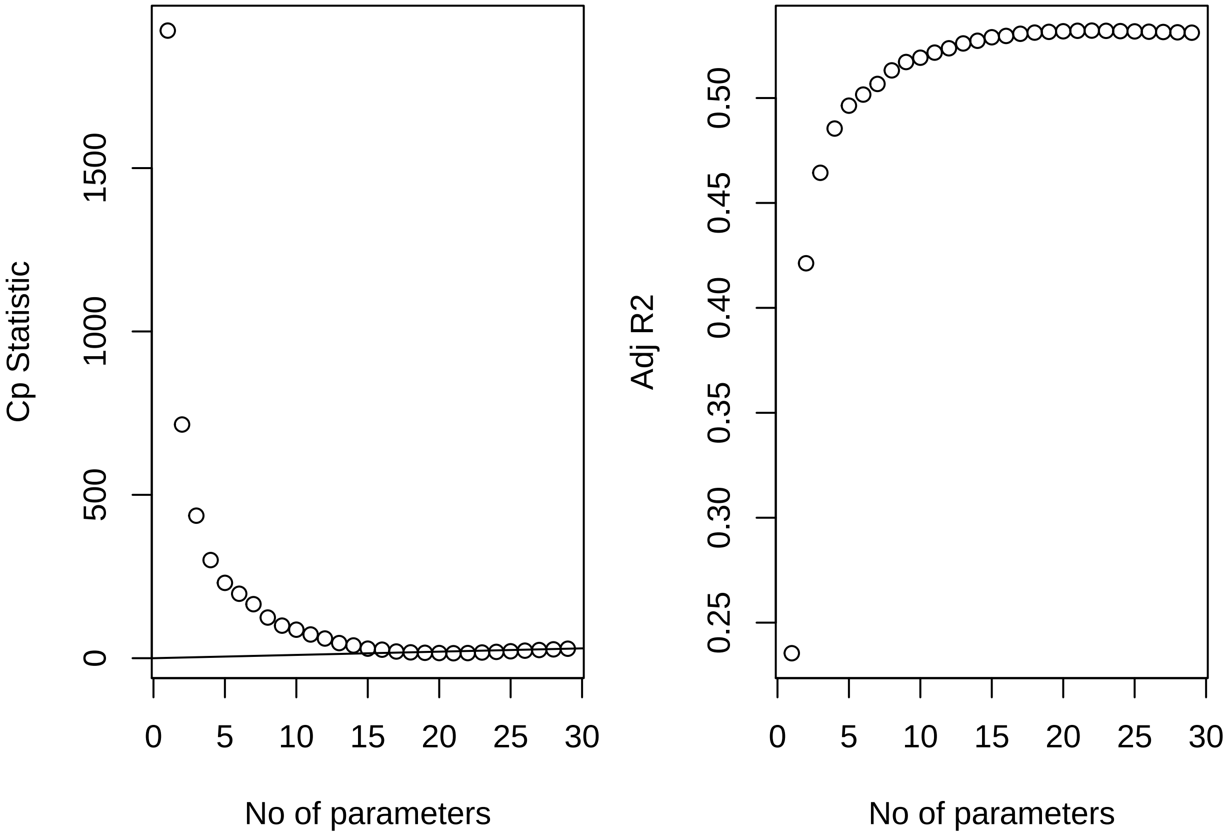
The Adjusted R^2 are very similar, doesn’t give too much information for model selection.

Plots of Cp and Adj-R2 as functions of parameters:

```
b = regsubsets(target_death_rate ~ ., data=data_vrb_pre_selec, nvmax = 30)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
## force.in = force.in, : 1 linear dependencies found

## Reordering variables and trying again:
```

```
rs = summary(b)

par(mar=c(4,4,1,1))
par(mfrow=c(1,2))

plot(1:29, rs$cp, xlab="No of parameters", ylab="Cp Statistic")
abline(0,1)

plot(1:29, rs$adjr2, xlab="No of parameters", ylab="Adj R2")
```



Summary of Model Selection

backwards: 16 variables. lm(formula = target_death_rate ~ incidence_rate + poverty_percent + median_age_female + percent_married + pct_hs18_24 + pct_hs25_over + pct_bach_deg25_over + pct_unemployed16_over + pct_private_coverage + pct_emp_priv_coverage + pct_public_coverage + pct_white + pct_black + pct_other_race + birth_rate + region, data = data_vrb_pre_selec) %>% summary()

forwards: 15 variables lm(formula = target_death_rate ~ pct_bach_deg25_over + incidence_rate + poverty_percent + pct_other_race + region + pct_public_coverage_alone + pct_hs25_over + pct_unemployed16_over + median_age_female + birth_rate + pct_some_col18_24 + pct_no_hs18_24 + pct_bach_deg18_24 + pct_private_coverage + pct_emp_priv_coverage, data = data_vrb_pre_selec) %>% summary stepwise: 19 variables. lm(formula = target_death_rate ~ avg_ann_count + incidence_rate + pop_est2015 + poverty_percent + median_age_female + percent_married + pct_no_hs18_24 + pct_hs18_24 + pct_hs25_over + pct_bach_deg25_over + pct_unemployed16_over + pct_private_coverage + pct_emp_priv_coverage + pct_public_coverage + pct_white + pct_black + pct_other_race + birth_rate + region, data = data_vrb_pre_selec) %>% summary()

Cp: 16 variables. lm(formula = target_death_rate ~ avg_ann_count + incidence_rate + pop_est2015 + poverty_percent + median_age_male + pct_no_hs18_24 + pct_hs18_24 + pct_hs25_over + pct_bach_deg25_over + pct_unemployed16_over + pct_private_coverage + pct_emp_priv_coverage + pct_public_coverage + pct_white + pct_other_race + birth_rate, data = data_vrb_pre_selec) %>% summary()

==

Variable Post-Selection (using collinearity and lit review)

we focus on: (1) Age (2) Education (3) Employment (4) Health Coverage

==
# Final Model Summary:

Original 33 varables:

(X)avg_ann_count: eliminated during post-selection, similar to 'incidence_rate' (X)avg_deaths_per_year: eliminated during pre-selection ~incidence_rate (X)med_income: eliminated during model selection (X)pop_est2015: eliminated during model selection ~poverty_percent (X)study_per_cap: eliminated during model selection (X)binned_inc: eliminated during pre-selection

(X)median_age: eliminated during model selection (X)median_age_male: eliminated during post-selection, correlated with 'median_age_female' ~median_age_female

~region(from original 'geography')

(X)avg_household_size: eliminated during model selection

(X)percent_married: eliminated during post-selection

(X)pct_no_hs18_24: eliminated during model selection ~pct_hs18_24 (X)pct_some_col18_24: eliminated during model selection (X)pct_bach_deg18_24: eliminated during model selection ~pct_hs25_over (X)pct_bach_deg25_over: eliminated during model selection

(X)pct_employed16_over: eliminated during pre-selection, has missing value ~pct_unemployed16_over

pct_private_coverage: eliminated during post-selection, lit review (X) pct_private_coverage_alone: eliminated during pre-selection, has missing value (X)pct_emp_priv_coverage: eliminated during post-selection, lit review ~pct_public_coverage (X) pct_public_coverage_alone eliminated during model selection

(X)pct_white: eliminated during post-selection (X)pct_black: eliminated during post-selection (X)pct_asian: eliminated during model selection (X)pct_other_race: eliminated during model selection

(X)pct_married_households: eliminated during during pre-selection (X)birth_rate: eliminated during during post-selection

Final: ################

```r
lm(formula = target_death_rate ~ incidence_rate + poverty_percent + median_age_female +
     region + pct_hs18_24 + pct_hs25_over + pct_unemployed16_over + pct_public_coverage,
     data = data_vrb_pre_selec) %>%
  summary(.)
```

```
##
## Call:
## lm(formula = target_death_rate ~ incidence_rate + poverty_percent +
##     median_age_female + region + pct_hs18_24 + pct_hs25_over +
##     pct_unemployed16_over + pct_public_coverage, data = data_vrb_pre_selec)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -115.435  -11.046    0.118   10.118  132.421
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           40.976245   4.800500   8.536  < 2e-16 ***
## incidence_rate         0.197127   0.006784  29.057  < 2e-16 ***
## poverty_percent        0.492955   0.107318   4.593 4.54e-06 ***
## median_age_female     -0.647972   0.103104  -6.285 3.76e-10 ***
## regionNortheast       -9.050428   1.481391  -6.109 1.13e-09 ***
## regionSouth            5.848026   0.927369   6.306 3.28e-10 ***
## regionWest            -9.693907   1.283539  -7.552 5.62e-14 ***
## pct_hs18_24            0.292025   0.045894   6.363 2.28e-10 ***
## pct_hs25_over          0.826709   0.066794  12.377  < 2e-16 ***
## pct_unemployed16_over  0.818317   0.144717   5.655 1.71e-08 ***
## pct_public_coverage    0.612658   0.095167   6.438 1.40e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.34 on 3036 degrees of freedom
## Multiple R-squared:  0.5162, Adjusted R-squared:  0.5146
## F-statistic: 323.9 on 10 and 3036 DF,  p-value: < 2.2e-16
```