bm_final_project

Adina, Henry, Haowei, Yun

December 6, 2018

Data cleaning

```
cancer = read_csv("./Cancer_Registry.csv") %>%
  janitor::clean_names() %>%
  separate(geography, into = c("county", "state"), sep = ",")

## demographic: age, race, marital status, income, education,
## employment and health insurance coverage

ses_cor = cancer %>%
  dplyr::select(med_income, poverty_percent, pct_unemployed16_over,
                pct_private_coverage, pct_emp_priv_coverage, pct_public_coverage,
                pct_public_coverage_alone)
corr = round(cor(ses_cor), 1)
ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE)
## income, employment status and health insurance coverage are highly correlated
cancer %>%
  dplyr::select(target_death_rate, med_income, poverty_percent) %>%
  ggscatmat() + geom_smooth(method = "lm")

income_edu = cancer %>%
  dplyr::select(target_death_rate, poverty_percent, pct_no_hs18_24, pct_hs18_24,
                pct_bach_deg18_24, pct_hs25_over, pct_bach_deg25_over)
corr = round(cor(income_edu), 1)
ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE)
## pct_hs25_over and pct_bach_deg25_over are highly correlated

## Check with missing values
skimr::skim(cancer %>% dplyr::select(-state, -county))

## Select variables
cancer = cancer %>%
  dplyr::select(state, county, target_death_rate, incidence_rate,
                study_per_cap, poverty_percent, median_age_female,
                percent_married, pct_no_hs18_24, pct_hs18_24,
                pct_bach_deg18_24, pct_hs25_over, pct_white,
                pct_asian, pct_other_race)

##  make histogram to see the distribution of continuous variables
cancer1 = cancer %>%
  dplyr::select(-state, -county)
cancer1 %>%
  gather(measure, value) %>%
  ggplot(aes(value)) +
  facet_wrap(. ~ measure, scales = "free") +
  geom_histogram()
```

Data exploration

```
## Distribution of continuous variables
descp_stats = function(x){
  df = broom::tidy(summary(x)) %>%
    mutate(sd = sd(x))
  return(df)
}

desp_var = cancer %>%
  dplyr::select(-state, -county) %>%
  map_df(descp_stats) %>%
  mutate(minimum = round(minimum, 2),
         q1 = round(q1, 2),
         median = round(median, 2),
         mean = round(mean, 2),
         q3 = round(q3, 2),
         maximum = round(maximum, 2),
         sd = round(sd, 2),
         variable = c("target_death_rate", "incidence_rate", "study_per_cap",
                      "poverty_percent", "median_age_female", "percent_married",
                      "pct_no_hs18_24", "pct_hs18_24", "pct_bach_deg18_24",
                      "pct_hs25_over", "pct_white", "pct_asian",
                      "pct_other_race")) %>%
  dplyr::select(variable, mean, sd, everything())
write_csv(desp_var, "decp_var.csv")
```

Fit the model

Stepwise regression

```
cancer_mod = cancer %>%
  dplyr::select(-state, -county)

fit = lm(target_death_rate ~ ., data = cancer_mod)
summary(fit)

step(fit, direction = "both")
refit = lm(target_death_rate ~ incidence_rate + poverty_percent +
    median_age_female + pct_hs18_24 + pct_bach_deg18_24 + pct_hs25_over +
    pct_white + pct_other_race, data = cancer_mod)
vif(refit) ## check multicollinearity
summary(refit)
par(mfrow = c(2,2))
plot(refit)
```

Criteria-based procedures

```
b = regsubsets(target_death_rate ~ ., data = cancer_mod, nvmax = 9)
   (rs = summary(b))

# Plots of Cp and Adj-R2 as functions of parameters
par(mar = c(4,4,1,1))
```

```r
par(mfrow = c(1,2))
plot(2:10, rs$cp, xlab = "No of parameters", ylab = "Cp Statistic")
abline(0,1)
plot(2:10, rs$adjr2, xlab = "No of parameters", ylab = "Adj R2")

refit_2 = lm(target_death_rate ~ incidence_rate + poverty_percent +
               pct_hs18_24+ pct_hs25_over, data = cancer_mod)
summary(refit_2)
par(mfrow = c(2,2))
plot(refit_2)

## check outliers
### outliers in Y
stu_res = rstandard(refit_2)
stu_res[abs(stu_res) > 2.5]

### leverage
hat_value = hatvalues(refit_2)
hat_value[hat_value > 0.2]

### influential points
influence.measures(refit_2)

## refit the model without potential outliers
mod_no = cancer_mod %>% slice(-282) %>% slice(-1220)
refit_no = lm(target_death_rate ~ incidence_rate + poverty_percent +
                pct_hs18_24+ pct_hs25_over, data = cancer_mod)
summary(refit_no)
```

Lasso regression

```r
Y <- as.matrix(cancer_mod[,1])
X <- as.matrix(cancer_mod[,-1])
grid <- 10^seq(5,-2, length=100)
ridge3<-glmnet(X, Y, alpha=1, lambda=grid)

set.seed(2)
cv.out<-cv.glmnet(X,Y)
plot(cv.out)
best.lambda<-cv.out$lambda.min
lasso2<-glmnet(X, Y, alpha =1, lambda=best.lambda)
coef(lasso2)

lasso2$dev.ratio
```

Cross validation

```r
## compare two models
library(caret)

## 8-predictor model
set.seed(2)
data_train<-trainControl(method="cv", number=5)
```

```
model_caret<-train(target_death_rate ~ incidence_rate + poverty_percent +
                     median_age_female + pct_hs18_24 + pct_bach_deg18_24 +
                     pct_hs25_over + pct_white + pct_other_race,
                   data = cancer_mod,
                   method='lm',
                   na.action=na.pass)

model_caret

## 4-predictor model
model_caret_2<-train(target_death_rate ~ incidence_rate + poverty_percent +
                       pct_hs18_24 + pct_hs25_over,
                   data = cancer_mod,
                   method='lm',
                   na.action=na.pass)

model_caret_2

## assess model predictive capability
library(MPV)                           # For PRESS criterion

newsummary <- function(model)
{
    list('coefs'    = round(t(summary(model)$coef[, 1:2]), 4),
         'criteria' = cbind('SSE'   = anova(model)["Residuals", "Sum Sq"],
                            'PRESS' = PRESS(model),
                            'MSE'   = anova(model)["Residuals", "Mean Sq"],
                            'Rsq'   = summary(model)$adj.r.squared))
}

newsummary(refit_2)
```