Data Manipulation, EDA, Statistical Learning Tools

Quinton Neville

12/8/2018

Load, clean, manipulate, and tidy the data

```
# Import data
cancer_raw = readr::read_csv("./Data/Cancer_Registry.csv") %>%
  janitor::clean_names()

dim(cancer_raw)
```

```
## [1] 3047   34
```

```
head(cancer_raw)
```

```
## # A tibble: 6 x 34
##   avg_ann_count avg_deaths_per_~ target_death_ra~ incidence_rate med_income
##           <dbl>            <int>            <dbl>          <dbl>      <int>
## 1          1397              469             165.           490.      61898
## 2           173               70             161.           412.      48127
## 3           102               50             175.           350.      49348
## 4           427              202             195.           430.      44243
## 5            57               26             144.           350.      49955
## 6           428              152             176            505.      52313
## # ... with 29 more variables: pop_est2015 <int>, poverty_percent <dbl>,
## #   study_per_cap <dbl>, binned_inc <chr>, median_age <dbl>,
## #   median_age_male <dbl>, median_age_female <dbl>, geography <chr>,
## #   avg_household_size <dbl>, percent_married <dbl>, pct_no_hs18_24 <dbl>,
## #   pct_hs18_24 <dbl>, pct_some_col18_24 <dbl>, pct_bach_deg18_24 <dbl>,
## #   pct_hs25_over <dbl>, pct_bach_deg25_over <dbl>,
## #   pct_employed16_over <dbl>, pct_unemployed16_over <dbl>,
## #   pct_private_coverage <dbl>, pct_private_coverage_alone <dbl>,
## #   pct_emp_priv_coverage <dbl>, pct_public_coverage <dbl>,
## #   pct_public_coverage_alone <dbl>, pct_white <dbl>, pct_black <dbl>,
## #   pct_asian <dbl>, pct_other_race <dbl>, pct_married_households <dbl>,
## #   birth_rate <dbl>
```

```
# Check NA values for each column
n_NA = sapply(cancer_raw[1:34], function(x) sum(length(which(is.na(x)))))
n_NA
```

```
##              avg_ann_count        avg_deaths_per_year
##                          0                          0
##          target_death_rate             incidence_rate
##                          0                          0
##                 med_income                pop_est2015
##                          0                          0
##            poverty_percent              study_per_cap
##                          0                          0
##                 binned_inc                 median_age
##                          0                          0
```

```
##            median_age_male          median_age_female
##                          0                          0
##                  geography         avg_household_size
##                          0                          0
##            percent_married            pct_no_hs18_24
##                          0                          0
##                pct_hs18_24         pct_some_col18_24
##                          0                       2285
##          pct_bach_deg18_24             pct_hs25_over
##                          0                          0
##        pct_bach_deg25_over        pct_employed16_over
##                          0                        152
##      pct_unemployed16_over       pct_private_coverage
##                          0                          0
## pct_private_coverage_alone      pct_emp_priv_coverage
##                        609                          0
##        pct_public_coverage  pct_public_coverage_alone
##                          0                          0
##                  pct_white                  pct_black
##                          0                          0
##                  pct_asian            pct_other_race
##                          0                          0
##     pct_married_households                birth_rate
##                          0                          0
```

```
# Check the percentage of NA values for each column
percentage_NA = sapply(cancer_raw[1:34], function(x) sum(length(which(is.na(x)))) / 3047)
percentage_NA %>% data.frame()
```

```
##                                .
## avg_ann_count             0.00000000
## avg_deaths_per_year       0.00000000
## target_death_rate         0.00000000
## incidence_rate            0.00000000
## med_income                0.00000000
## pop_est2015               0.00000000
## poverty_percent           0.00000000
## study_per_cap             0.00000000
## binned_inc                0.00000000
## median_age                0.00000000
## median_age_male           0.00000000
## median_age_female         0.00000000
## geography                 0.00000000
## avg_household_size        0.00000000
## percent_married           0.00000000
## pct_no_hs18_24            0.00000000
## pct_hs18_24               0.00000000
## pct_some_col18_24         0.74991795
## pct_bach_deg18_24         0.00000000
## pct_hs25_over             0.00000000
## pct_bach_deg25_over       0.00000000
## pct_employed16_over       0.04988513
## pct_unemployed16_over     0.00000000
## pct_private_coverage      0.00000000
## pct_private_coverage_alone 0.19986872
```

```
## pct_emp_priv_coverage      0.00000000
## pct_public_coverage        0.00000000
## pct_public_coverage_alone  0.00000000
## pct_white                  0.00000000
## pct_black                  0.00000000
## pct_asian                  0.00000000
## pct_other_race             0.00000000
## pct_married_households     0.00000000
## birth_rate                 0.00000000
```

```
#Pulling quartiles for study_per_cap categorical manipulation
study.quart <- with(cancer_raw, study_per_cap[study_per_cap > 0]) %>%
  quantile(., probs = c(0.25, 0.5, 0.75))

#Variable Manipulation
cancer.df <- cancer_raw %>%    #Remove Rows with > 20% missing
  dplyr::select(-pct_some_col18_24) %>%  #Remove for too many missing
  mutate(
    pct_non_white = pct_black + pct_asian + pct_other_race, #Creating white, non-white percentages vari
    state = str_split_fixed(geography, ", ", 2)[ ,2] %>% as.factor(), #pulling state variable and casti
    binned_inc_lb = str_split_fixed(binned_inc, ", ", 2)[ ,1] %>% parse_number(), #pulling numeric lowe
    binned_inc_ub = str_split_fixed(binned_inc, ", ", 2)[ ,2] %>% parse_number(), #pulling numeric uppe
    binned_inc_point = (binned_inc_lb + binned_inc_ub)/2, #computing point estimate from ub,lb (interva
    study_quantile = ifelse(study_per_cap == 0, "None",
                            ifelse(study_per_cap > 0 & study_per_cap <= study.quart[1], "Low",
                                   ifelse(study_per_cap > study.quart[1] & study_per_cap <= study.quart[
                                          ifelse(study_per_cap > study.quart[2] & study_per_cap <= study
                                                 "Very High")))),
    study_quantile = as.factor(study_quantile) %>% fct_relevel(., "None", "Low", "Moderate", "High", "V
    avg_deaths_yr_pop = avg_deaths_per_year/pop_est2015,  #incorporate two vars into one (multicollinea
    avg_ann_count_pop = avg_ann_count/pop_est2015 #incorporate two vars into one (multicollinearity)
  ) %>%
  dplyr::select(-c(binned_inc, geography, study_per_cap))
```

Imputing Values with less than 20% missing (two variables)

- pct_employed16_over ~ 4%
- pct_private_coverage_alone ~ 20%

```
library(glmnet)
library(tidyverse)
#Impute those missing less than 20%
#1. pct_employed16_over
#2. pct_private_coverage_alone

#Set up appropriate test and train for pct_employed16_over (removing other missing % variable and respo
train.df <- cancer.df %>% dplyr::select(-c(pct_private_coverage_alone, target_death_rate)) %>% filter(!
test.df <- cancer.df %>% dplyr::select(-c(pct_private_coverage_alone, target_death_rate)) %>% filter(is

#Set up Matrices
#Create Design Matrix Train
X <- train.df %>%
  dplyr::select(-pct_employed16_over) %>%
```

```
  names() %>%
  paste("~ ", paste(., collapse = "+")) %>%
  formula() %>%
  model.matrix(.,train.df)

#Create Design Matrix Test
X1 <- test.df %>%
  dplyr::select(-pct_employed16_over) %>%
  names() %>%
  paste("~ ", paste(., collapse = "+")) %>%
  formula() %>%
  model.matrix(., test.df)

#Remove Intercept
X <- X[,-1]
X1 <- X1[,-1]

#Create Response vector (as matrix)
Y <- train.df %>% dplyr::select(pct_employed16_over) %>% as.matrix()

#Optimize lambda
lambda.grid <- 10^seq(-3,1,length = 100)

#CV n = 10
cv.lasso <- cv.glmnet(X, Y, alpha = 1, intercept = TRUE, lambda = lambda.grid, family = "gaussian")

#Grab optimal lambda
opt.lambda.lasso <- cv.lasso$lambda.min

#Run model
unemploy.lasso <- glmnet(X, Y, alpha = 1, intercept = TRUE, lambda = opt.lambda.lasso, family = "gaussi

#Impute employed16_over_preds (first since it has less missing data ~4%)
employed16_over_preds <- predict(unemploy.lasso, newx = X1)

#Set up appropriate test and train
train.df <- cancer.df %>% dplyr::select(-c(pct_employed16_over, target_death_rate)) %>% filter(!is.na(p
test.df <- cancer.df %>% dplyr::select(-c(pct_employed16_over, target_death_rate)) %>% filter(is.na(pct_

#Set up Matrices
#Create Design Matrix Train
X <- train.df %>%
  dplyr::select(-pct_private_coverage_alone) %>%
  names() %>%
  paste("~ ", paste(., collapse = "+")) %>%
  formula() %>%
  model.matrix(.,train.df)

#Create Design Matrix Test
X1 <- test.df %>%
  dplyr::select(-pct_private_coverage_alone) %>%
  names() %>%
  paste("~ ", paste(., collapse = "+")) %>%
```

```
  formula() %>%
  model.matrix(., test.df)

#Remove Intercept
X <- X[,-1]
X1 <- X1[,-1]

#Create Response vector (as matrix)
Y <- train.df %>% dplyr::select(pct_private_coverage_alone) %>% as.matrix()

#Optimize lambda
lambda.grid <- 10^seq(-3,1,length = 100)

#CV n = 10
cv.lasso <- cv.glmnet(X, Y, alpha = 1, intercept = TRUE, lambda = lambda.grid, family = "gaussian")

#Grab optimal lambda
opt.lambda.lasso <- cv.lasso$lambda.min

#Run model
cov.lasso <- glmnet(X, Y, alpha = 1, intercept = TRUE, lambda = opt.lambda.lasso, family = "gaussian")

#Impute pct_private_coverage_alone (second since it has more missing data ~20%)
pct_private_coverage_alone_preds <- predict(cov.lasso, newx = X1)

#Replace Imputed values
cancer.df <- cancer.df %>%
  mutate(imp_pct_employed16_over = ifelse(is.na(pct_employed16_over),
                                          employed16_over_preds, pct_employed16_over),
         imp_pct_private_coverage_alone = ifelse(is.na(pct_private_coverage_alone),
                                          pct_private_coverage_alone_preds, pct_private_coverage_alone)
        )

#Check
verif.df <- cancer.df %>%
  dplyr::select(pct_employed16_over, imp_pct_employed16_over, pct_private_coverage_alone, imp_pct_priva
```

looks good so we will take out extraneous variables for final df.

```
#Looks good, so we will replace for our final data set
cancer.df <- cancer.df %>%
  dplyr::select(-c(pct_employed16_over, pct_private_coverage_alone))

#Check it out
str(cancer.df)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    3047 obs. of  38 variables:
##  $ avg_ann_count                : num  1397 173 102 427 57 ...
##  $ avg_deaths_per_year          : int  469 70 50 202 26 152 97 71 36 1380 ...
##  $ target_death_rate            : num  165 161 175 195 144 ...
##  $ incidence_rate               : num  490 412 350 430 350 ...
##  $ med_income                   : int  61898 48127 49348 44243 49955 52313 37782 40189 42579 60397
##  $ pop_est2015                  : int  260131 43269 21026 75882 10321 61023 41516 20848 13088 84395
##  $ poverty_percent              : num  11.2 18.6 14.6 17.1 12.5 15.6 23.2 17.8 22.3 13.1 ...
##  $ median_age                   : num  39.3 33 45 42.8 48.3 45.4 42.6 51.7 49.3 35.8 ...
```

```
##  $ median_age_male               : num  36.9 32.2 44 42.2 47.8 43.5 42.2 50.8 48.4 34.7 ...
##  $ median_age_female             : num  41.7 33.7 45.8 43.4 48.9 48 43.5 52.5 49.8 37 ...
##  $ avg_household_size            : num  2.54 2.34 2.62 2.52 2.34 2.58 2.42 2.24 2.38 2.65 ...
##  $ percent_married               : num  52.5 44.5 54.2 52.7 57.8 50.4 54.1 52.7 55.9 50 ...
##  $ pct_no_hs18_24                : num  11.5 6.1 24 20.2 14.9 29.9 26.1 27.3 34.7 15.6 ...
##  $ pct_hs18_24                   : num  39.5 22.4 36.6 41.2 43 35.1 41.4 33.9 39.4 36.3 ...
##  $ pct_bach_deg18_24             : num  6.9 7.5 9.5 2.5 2 4.5 5.8 2.2 1.4 7.1 ...
##  $ pct_hs25_over                 : num  23.2 26 29 31.6 33.4 30.4 29.8 31.6 32.2 28.8 ...
##  $ pct_bach_deg25_over           : num  19.6 22.7 16 9.3 15 11.9 11.9 11.3 12 16.2 ...
##  $ pct_unemployed16_over         : num  8 7.8 7 12.1 4.8 12.9 8.9 8.9 10.3 9.2 ...
##  $ pct_private_coverage          : num  75.1 70.2 63.7 58.4 61.6 60 49.5 55.8 55.5 69.9 ...
##  $ pct_emp_priv_coverage         : num  41.6 43.6 34.9 35 35.1 32.6 28.3 25.9 29.9 44.4 ...
##  $ pct_public_coverage           : num  32.9 31.1 42.1 45.3 44 43.2 46.4 50.9 48.1 31.4 ...
##  $ pct_public_coverage_alone     : num  14 15.3 21.1 25 22.7 20.2 28.7 24.1 26.6 16.5 ...
##  $ pct_white                     : num  81.8 89.2 90.9 91.7 94.1 ...
##  $ pct_black                     : num  2.595 0.969 0.74 0.783 0.27 ...
##  $ pct_asian                     : num  4.822 2.246 0.466 1.161 0.666 ...
##  $ pct_other_race                : num  1.843 3.741 2.747 1.363 0.492 ...
##  $ pct_married_households        : num  52.9 45.4 54.4 51 54 ...
##  $ birth_rate                    : num  6.12 4.33 3.73 4.6 6.8 ...
##  $ pct_non_white                 : num  9.26 6.96 3.95 3.31 1.43 ...
##  $ state                         : Factor w/ 51 levels "Alabama","Alaska",..: 48 48 48 48 48 48 48 48
##  $ binned_inc_lb                 : num  61495 48022 48022 42724 48022 ...
##  $ binned_inc_ub                 : num  125635 51046 51046 45201 51046 ...
##  $ binned_inc_point              : num  93565 49534 49534 43963 49534 ...
##  $ study_quantile                : Factor w/ 5 levels "None","Low","Moderate",..: 5 2 2 4 1 4 1 1 1 4
##  $ avg_deaths_yr_pop             : num  0.0018 0.00162 0.00238 0.00266 0.00252 ...
##  $ avg_ann_count_pop             : num  0.00537 0.004 0.00485 0.00563 0.00552 ...
##  $ imp_pct_employed16_over       : num  51.9 55.9 45.9 48.3 48.2 44.1 51.8 40.9 39.5 56.6 ...
##  $ imp_pct_private_coverage_alone: num  54.7 53.8 43.5 40.3 43.9 ...
```

```r
dim(cancer.df)
```

```
## [1] 3047   38
```

```r
#Check new percentage missing after removing one and imputing two
# Check the percentage of NA values for each column
percentage_NA = apply(cancer.df, 2, function(x) sum(length(which(is.na(x)))) / nrow(cancer.df))
percentage_NA %>% data.frame() %>% knitr::kable()
```

	.
avg_ann_count	0
avg_deaths_per_year	0
target_death_rate	0
incidence_rate	0
med_income	0
pop_est2015	0
poverty_percent	0
median_age	0
median_age_male	0
median_age_female	0
avg_household_size	0
percent_married	0
pct_no_hs18_24	0
pct_hs18_24	0

	.
pct_bach_deg18_24	0
pct_hs25_over	0
pct_bach_deg25_over	0
pct_unemployed16_over	0
pct_private_coverage	0
pct_emp_priv_coverage	0
pct_public_coverage	0
pct_public_coverage_alone	0
pct_white	0
pct_black	0
pct_asian	0
pct_other_race	0
pct_married_households	0
birth_rate	0
pct_non_white	0
state	0
binned_inc_lb	0
binned_inc_ub	0
binned_inc_point	0
study_quantile	0
avg_deaths_yr_pop	0
avg_ann_count_pop	0
imp_pct_employed16_over	0
imp_pct_private_coverage_alone	0

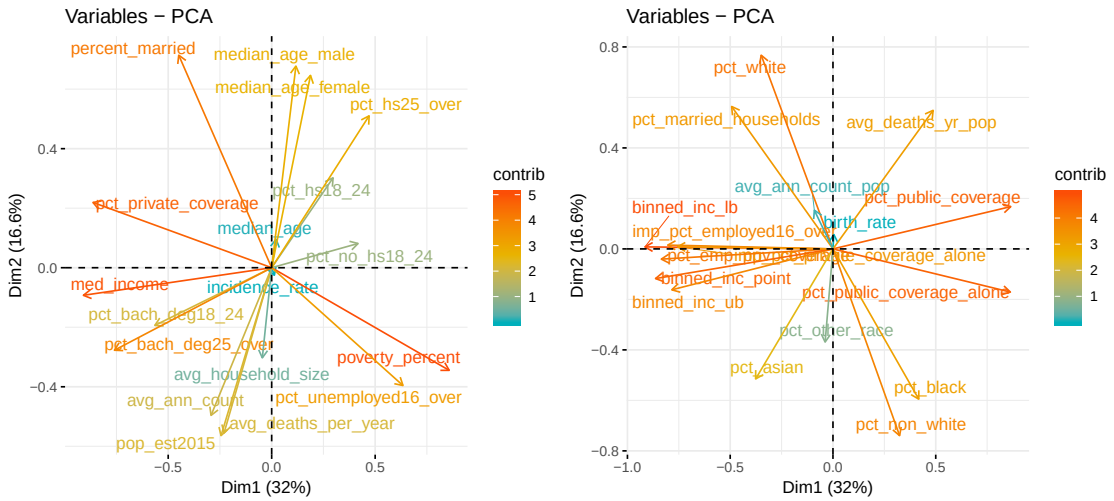
```
#No more missing data and we only had to throw out one variable
```

PCA Analysis for Variable Selection

*Plot is a bit messy, take subsets of the data and repeat

```
#Scale and perform pca (take out non-continuous vars)
cancer.pca <- cancer.df %>%
  dplyr::select(-c(state, study_quantile, target_death_rate)) %>%
  scale() %>%
  as.data.frame() %>%
  prcomp()


#str(cancer.pca)

name.vec <- cancer.df %>% dplyr::select(-c(state, study_quantile, target_death_rate)) %>% names()

pca.viz1 <- fviz_pca_var(cancer.pca,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE,
             select.var = list(name = name.vec[1:18]))

pca.viz2 <- fviz_pca_var(cancer.pca,
```

```
            col.var = "contrib",
            gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
            repel = TRUE,
            select.var = list(name = name.vec[19:35]))

pca.viz1 + pca.viz2
```



Here, we used PCA component analysis on the scaled continuous predictor set, projecting our $p = 35$ dimensional predictor space onto the 2 dimensional PCA space with the two principal component vectors explaining the highest % of variability in the data(basically the best basis in $\mathcal{R}^{\in}$to visualize how related each of our continuous predictors are to each other) vectors pointing in the same directions are explaining very simimlar types of the variance in the data (i.e. related, potentially multicollinearity), and the length(color) describes the magnitude or strength of how much of the variance in the data the predictor is explaining. It takes a while staring at it to understand exactly what is going on, and I split into two plots for clarity. Here are my takeaways:

1. Median age female and male explain a good amount of variability but are very related, we should take a average of the two for an average median age
2. Percent white and percent married are explaining similiar variability at similiar strength, pct races are explaining different types of variability in the data, while pct_white and pct_non_white are explaining inverse types of variability in the data (180 degree angle) makes me think we should keep seperate race percentages or pct_non_white, but not both (obviously) one while be better than the other.
3. Avg_ann_count, avg deaths, pop_est2015 are all explaining the same type and proportion of variability, should only use one (my best guess is pop_est2015 based on magnitude.)
4. pct_private, pct_public_cov, and pct_public alone are all explaining different types of variability but at sufficient magnitude and should be kept. However pct_private_cov is highly correlated with income, and should not be included if any type of income variable is in the model.
5. median_age is not a strong explanotary variable (by magnitude), and weirdly does not equal median_male + median_female /2. So I say we lose median_age and keep a variable for avg_median = median_male + median_female /2.
6. Drop the mutate avg_ann_count_pop, low magnitude and not explaining anything significantly. See (3.) for reccomendation on which var to select there.
7. For income, all median_income, binned_inc_point estimate, binned_inc_lb, and binned_ub very related. I think we should either use median_income OR binned_inc_point estimate, but retain the lb, and ub as they seem to be explained different types of variability at good magnitude.

8. pct_bach_deg25_over is better than pecent_bach_deg18_24, explaining similar variability. Maybe take an average of the two or only include pct_bach_deg25_over.
9. Incidence rate has relatively small magnitude, but in a direction almost no other variable takes, so that should be kept I think.
10.