APPENDIX

OVERVIEW

The dataset used for this project is 'Cancer mortality rates in the US' for the year 2010-2015 at county level. This includes the data for 50 states and the District of Colombia. Based on this data, we want to build a model that estimates the cancer death rates in the US and try to predict the risk that a person from that county might be suffering from cancer. The data dictionary is as shown below:

Column	Description
TARGET_deathRate	Dependent variable. Mean per capita (100,000) cancer mortalities
avgAnnCount	Mean number of reported cases of cancer diagnosed annually
avgDeathsPerYear	Mean number of reported mortalities due to cancer
incidenceRate	Mean per capita (100,000) cancer diagoses
medianIncome	Median income per county
popEst2015	Population of county
povertyPercent	Percent of populace in poverty
studyPerCap	Per capita number of cancer-related clinical trials per county
binnedInc	Median income per capita binned by decile
MedianAge	Median age of county residents
MedianAgeMale	Median age of male county residents
MedianAgeFemale	Median age of female county residents
Geography	County name
AvgHouseholdSize	Mean household size of county
PercentMarried	Percent of county residents who are married
PctNoHS18_24	Percent of county residents ages 18-24 highest education attained, less than high school
PctHS18_24	Percent of county residents ages 18-24 highest education attained, high school diploma
PctSomeCol18_24	Percent of county residents ages 18-24 highest education attained, some college
PctBachDeg18_24	Percent of county residents ages 18-24 highest education attained, bachelor's degree
PctHS25_Over	Percent of county residents ages 25 and over highest education attained, high school diploma
PctBachDeg25_Over	Percent of county residents ages 25 and over highest education attained
PctEmployed16_Over	Percent of county residents ages 16 and over employed
PctUnemployed16_Over	Percent of county residents ages 16 and over unemployed
PctPrivateCoverage	Percent of county residents with private health coverage
PctPrivateCoverageAlone	Percent of county residents with private health coverage alone (no public assistance)
PctEmpPrivCoverage	Percent of county residents with employee-provided private health coverage
PctPublicCoverage	Percent of county residents with government-provided health coverage
PctPubliceCoverageAlone	Percent of county residents with government-provided health coverage alone
PctWhite	Percent of county residents who identify as White
PctBlack	Percent of county residents who identify as Black
PctAsian	Percent of county residents who identify as Asian
PctOtherRace	Percent of county residents who identify in a category which is not White, Black, or Asian
PctMarriedHouseholds	Percent of married households
BirthRate	Number of live births relative to number of women in county

EXPLORATORY DATA ANALYSIS:

Exploratory analysis is often the initial step of data analysis. Here we get familiar with data, visualize the data in a number of forms, analyze relationships between the variables, look for outliers, patterns, and trends in the data.

The very first step in EDA is to check the dimension of the input dataset and the type of variables.

```
> str(Canc)
'data.frame':   3047 obs. of  34 variables:
 $ avgAnnCount            : num  1397 173 102 427 57 ...
 $ avgDeathsPerYear       : int  469 70 50 202 26 152 97 71 36 1380 ...
 $ TARGET_deathRate       : num  165 161 175 195 144 ...
 $ incidenceRate          : num  490 412 350 430 350 ...
 $ medIncome              : int  61898 48127 49348 44243 49955 52313 37782 40189 42579 60397 ...
 $ popEst2015             : int  260131 43269 21026 75882 10321 61023 41516 20848 13088 843954 ...
 $ povertyPercent         : num  11.2 18.6 14.6 17.1 12.5 15.6 23.2 17.8 22.3 13.1 ...
 $ studyPerCap            : num  499.7 23.1 47.6 342.6 0 ...
 $ binnedInc              : Factor w/ 10 levels "(34218.1, 37413.8]",..: 9 6 6 4 6 7 2 2 3 8 ...
 $ MedianAge              : num  39.3 33 45 42.8 48.3 45.4 42.6 51.7 49.3 35.8 ...
 $ MedianAgeMale          : num  36.9 32.2 44 42.2 47.8 43.5 42.2 50.8 48.4 34.7 ...
 $ MedianAgeFemale        : num  41.7 33.7 45.8 43.4 48.9 48 43.5 52.5 49.8 37 ...
 $ Geography              : Factor w/ 3047 levels "Abbeville County, South Carolina",..: 1459 1460 1464 1589 1618 1
766 2051 2112 2143 2185 ...
 $ AvgHouseholdSize       : num  2.54 2.34 2.62 2.52 2.34 2.58 2.42 2.24 2.38 2.65 ...
 $ PercentMarried         : num  52.5 44.5 54.2 52.7 57.8 50.4 54.1 52.7 55.9 50 ...
 $ PctNoHS18_24           : num  11.5 6.1 24 20.2 14.9 29.9 26.1 27.3 34.7 15.6 ...
 $ PctHS18_24             : num  39.5 22.4 36.6 41.2 43 35.1 41.4 33.9 39.4 36.3 ...
 $ PctSomeCol18_24        : num  42.1 64 NA 36.1 40 NA NA 36.5 NA NA ...
 $ PctBachDeg18_24        : num  6.9 7.5 9.5 2.5 2 4.5 5.8 2.2 1.4 7.1 ...
 $ PctHS25_Over           : num  23.2 26 29 31.6 33.4 30.4 29.8 31.6 32.2 28.8 ...
 $ PctBachDeg25_Over      : num  19.6 22.7 16 9.3 15 11.9 11.9 11.3 12 16.2 ...
 $ PctEmployed16_Over     : num  51.9 55.9 45.9 48.3 48.2 44.1 51.8 40.9 39.5 56.6 ...
 $ PctUnemployed16_Over   : num  8 7.8 7 12.1 4.8 12.9 8.9 8.9 10.3 9.2 ...
 $ PctPrivateCoverage     : num  75.1 70.2 63.7 58.4 61.6 60 49.5 55.8 55.5 69.9 ...
 $ PctPrivateCoverageAlone: num  NA 53.8 43.5 40.3 43.9 38.8 35 33.1 37.8 NA ...
 $ PctEmpPrivCoverage     : num  41.6 43.6 34.9 35 35.1 32.6 28.3 25.9 29.9 44.4 ...
 $ PctPublicCoverage      : num  32.9 31.1 42.1 45.3 44 43.2 46.4 50.9 48.1 31.4 ...
 $ PctPublicCoverageAlone : num  14 15.3 21.1 25 22.7 20.2 28.7 24.1 26.6 16.5 ...
 $ PctWhite               : num  81.8 89.2 90.9 91.7 94.1 ...
 $ PctBlack               : num  2.595 0.969 0.74 0.783 0.27 ...
 $ PctAsian               : num  4.822 2.246 0.466 1.161 0.666 ...
 $ PctOtherRace           : num  1.843 3.741 2.747 1.363 0.492 ...
 $ PctMarriedHouseholds   : num  52.9 45.4 54.4 51 54 ...
 $ BirthRate              : num  6.12 4.33 3.73 4.6 6.8 ...
```

Figure 1

From Figure 1 we can sight that the dataset contains 3047 rows and 34 columns. Majority of the variables are numeric with two categorical variables that are Geography and binned Income.

DATA CLEANING AND PRE-PROCESSING

Imputation of missing values

Our second step in the EDA is to perform some pre-processing and check if the given input data has any missing values, before diving deep into the analysis.

```
> df_status(Canc)
                    variable q_zeros p_zeros q_na  p_na q_inf p_inf    type unique
1               avgAnnCount       0    0.00    0  0.00     0     0 numeric    929
2          avgDeathsPerYear       0    0.00    0  0.00     0     0 integer    608
3          TARGET_deathRate       0    0.00    0  0.00     0     0 numeric   1053
4             incidenceRate       0    0.00    0  0.00     0     0 numeric   1506
5                 medIncome       0    0.00    0  0.00     0     0 integer   2920
6                 popEst2015       0    0.00    0  0.00     0     0 integer   2999
7           povertyPercent       0    0.00    0  0.00     0     0 numeric    333
8              studyPerCap    1931   63.37    0  0.00     0     0 numeric   1117
9                binnedInc       0    0.00    0  0.00     0     0  factor     10
10                MedianAge       0    0.00    0  0.00     0     0 numeric    325
11          MedianAgeMale       0    0.00    0  0.00     0     0 numeric    298
12        MedianAgeFemale       0    0.00    0  0.00     0     0 numeric    296
13              Geography       0    0.00    0  0.00     0     0  factor   3047
14       AvgHouseholdSize       0    0.00    0  0.00     0     0 numeric    199
15         PercentMarried       0    0.00    0  0.00     0     0 numeric    362
16          PctNoHS18_24       2    0.07    0  0.00     0     0 numeric    405
17            PctHS18_24       1    0.03    0  0.00     0     0 numeric    469
18        PctSomeCol18_24       0    0.00 2285 74.99     0     0 numeric    343
19        PctBachDeg18_24     118    3.87    0  0.00     0     0 numeric    219
20          PctHS25_Over       0    0.00    0  0.00     0     0 numeric    361
21      PctBachDeg25_Over       0    0.00    0  0.00     0     0 numeric    281
22     PctEmployed16_Over       0    0.00  152  4.99     0     0 numeric    409
23   PctUnemployed16_Over       0    0.00    0  0.00     0     0 numeric    195
24     PctPrivateCoverage       0    0.00    0  0.00     0     0 numeric    498
25 PctPrivateCoverageAlone       0    0.00  609 19.99     0     0 numeric    459
26     PctEmpPrivCoverage       0    0.00    0  0.00     0     0 numeric    450
27      PctPublicCoverage       0    0.00    0  0.00     0     0 numeric    395
28  PctPublicCoverageAlone       0    0.00    0  0.00     0     0 numeric    319
29               PctWhite       0    0.00    0  0.00     0     0 numeric   3044
30               PctBlack      72    2.36    0  0.00     0     0 numeric   2972
31               PctAsian     194    6.37    0  0.00     0     0 numeric   2852
32           PctOtherRace     140    4.59    0  0.00     0     0 numeric   2903
33   PctMarriedHouseholds       0    0.00    0  0.00     0     0 numeric   3043
34              BirthRate       4    0.13    0  0.00     0     0 numeric   3019
```

Figure 2

Figure 2 shows various features such as the quantity of zeros(q_zeros), percentage of zeros(p_zeros), quantity of infinite values (q_inf), percentage of infinite values(p_inf), quantity of NA(q_na) percentage of NA(p_na), data type (type), quantity of unique values (unique) for every column in the data set file. The reason for considering this output is that variables with lots of zeros, several missing values may not be useful for analysis and can cause a bias model.We can infer that there are missing values in three columns of the data set - PctSomeCol18_24 (74.99%), PctEmployed16_Over (4.99%) and PctPrivateCoverageAlone (19.99%). Since column PctSomeCol18_24 has very high number of missing values and we could not locate the missing values, we will drop this column. For the remaining two columns performed imputation of missing values with mean as mean and median both shows the same value.

Handling categorical variable

The geography column contains two information in it. Therefore, split the geography column into two columns as county and state. This can be viewed in Figure 3.

```
> str(Canc$county)
 Factor w/ 1819 levels "Abbeville County",..: 876 877 881 946 951 1013 1190 1230 1254 1275 ...
> str(Canc$state)
 Factor w/ 51 levels "Alabama","Alaska",..: 48 48 48 48 48 48 48 48 48 48 ...
```

Figure 3

There are three categorical columns in this data set - binnedInc, county and state.

Every observation in this data set corresponds to one county since the data is collected by county. Hence there are 3047 unique county values. The state variable contains 51 states of the US country. The binnedInc has 10 levels and this variable has already been divided into brackets for better interpretation of the median income per capita binned by decile.

Outlier detection and outlier handling

All the variables in this dataset have outliers. We analyzed every variable through box plots and inferred that actions need to be performed on median age which contained values higher than 200 and in real these are impossible numbers for any person to live so long. Hence, removed the outliers in median age variable which were above 200 that contributed to 89 observations of 3047. Next, the target variable death rate had 64 outliers which did not seem valid numbers and we dropped them. The final dataset contains 2894 observations of 34 variables.

```
> print(outlier_MedianAge[outlier_MedianAge>200])
 [1] 458.4 469.2 546.0 624.0 508.8 619.2 498.0 412.8 481.2 424.8 535.2 406.8 579.6 502.8 496.8 525.6
[17] 519.6 536.4 523.2 470.4 430.8 414.0 500.4 429.6 496.8 349.2 511.2 498.0 508.8
> length(outlier_MedianAge)
[1] 89
```

s

In the next section, we will explore all the variables and get a good understanding of each variable before building a model. We will look at the histogram which tells us about the shape of the distributions, box plot to see if there are any potential outliers and summary statistics for each of the numerical variables.

Univariate analysis

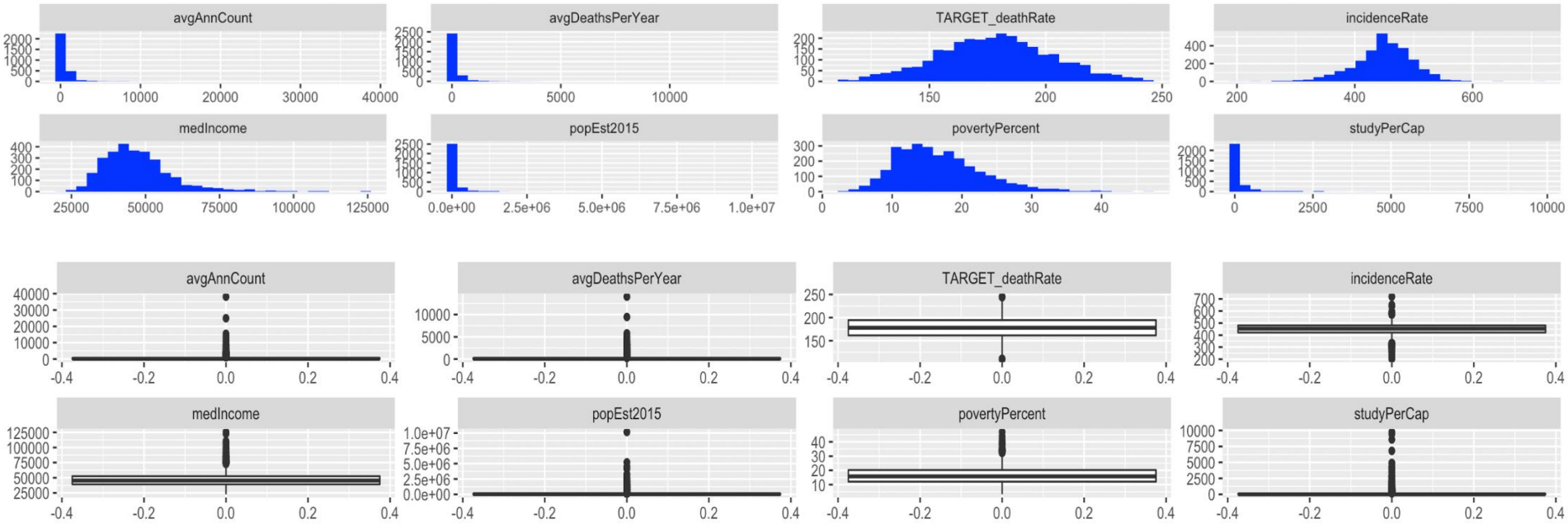


Figure 4

From the univariate graphs above, we can infer the following:

- The average number of cancer cases diagnosed annually is a highly right skewed and the median count of 176 person. (avgAnnCount)
- The average number of reported deaths due to cancer is also a right skewed with the median deaths being 63 per year. (avgDeathsPerYear)
- The mean number of cancer death rate out of 100,000 people per year by county is 178.0. (TARGET_deathRate)
- The rate at which people get cancer of 100,000 people by county shows a median value of 453.5 and the incidenceRate variable distribution is slight right skewed. (incidenceRate)
- The median household income of the county is $45331. (medIncome)
- The median estimated population of the number of people living in a county is 27329 and this data is highly right skewed. (popEst2015)
- The percent of poverty by county shows an estimated mean of 16 percent. (povertyPercent)
- The mean number of cancer-related clinical trials per capita (per county) is 155 and this data is highly right skewed. (studyPerCap)

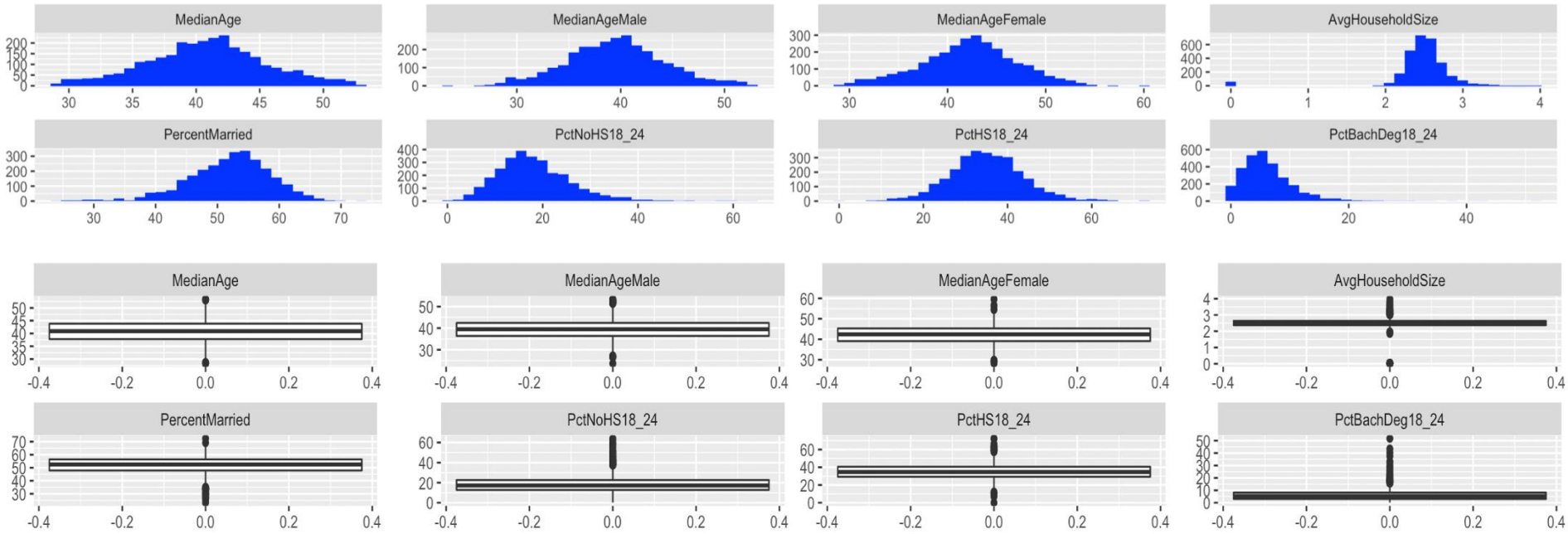


Figure 5

From the univariate graphs above, we can infer the following:

- The mean median age of the county residents is 45 years which means that half the people younger than age 45 and half are older and this variable is normally distributed. (MedianAge)
- The mean median age of male county residents is 39 years and this variable has a normal distribution. (MedianAgeMale)
- The mean median age of female county residents is 42 years and this data is normally distributed. (MedianAgeFemale)
- The average persons per household shows a mean of 2.5 people. (AvgHouseholdSize)
- 50 percent of county residents are married, and this variable has a slight normal distribution. (PercentMarried)
- 17.10 percent of county residents ages 18-24 highest education attained are less than high school. (PctNoHS18_24)
- 34.93 percent of county residents ages 18-24 highest education attained are high school diploma. (PctHS18_24)
- 5.4 percent of the county residents ages 18-24 highest education attained are bachelor's degree. (PctBachDeg18_24)

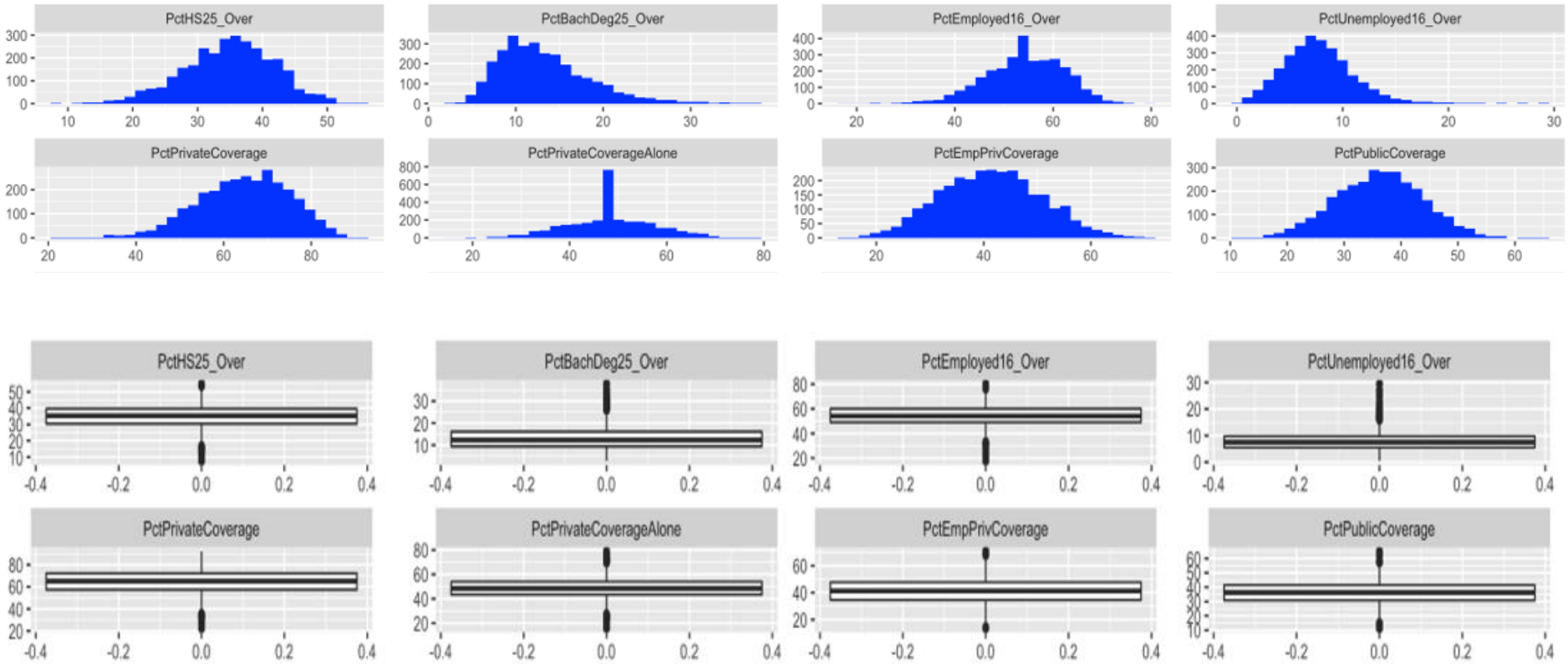


Figure 6

From the univariate graphs above, we can infer the following:

- 35 percent of county residents ages 25 and above highest education attained high school diploma, this variable has a slight normal distribution. (PctHS25_Over)
- 12.3 percent of county residents ages 25 and above highest education attained: bachelor's degree. (PctBachDeg25_Over)
- There are 54.26 percent of county residents ages 16 and above employed, this data is normally distributed. (PctEmployed16_Over)
- 7.8 percent of county residents ages 16 and over unemployed, this data is normally distributed. (PctUnemployed16_Over)
- 64.5 percent of county residents have private health coverage. (PctPrivateCoverage)
- 48.57 percent of county residents are with private health coverage alone and no public assistance. (PctPrivateCoverageAlone)
- 41.48 percent of county residents are with employee-provided private health coverage. (PctEmpPrivCoverage)
- There are 36.15 percent of county residents with government-provided health coverage. (PctPublicCoverage)

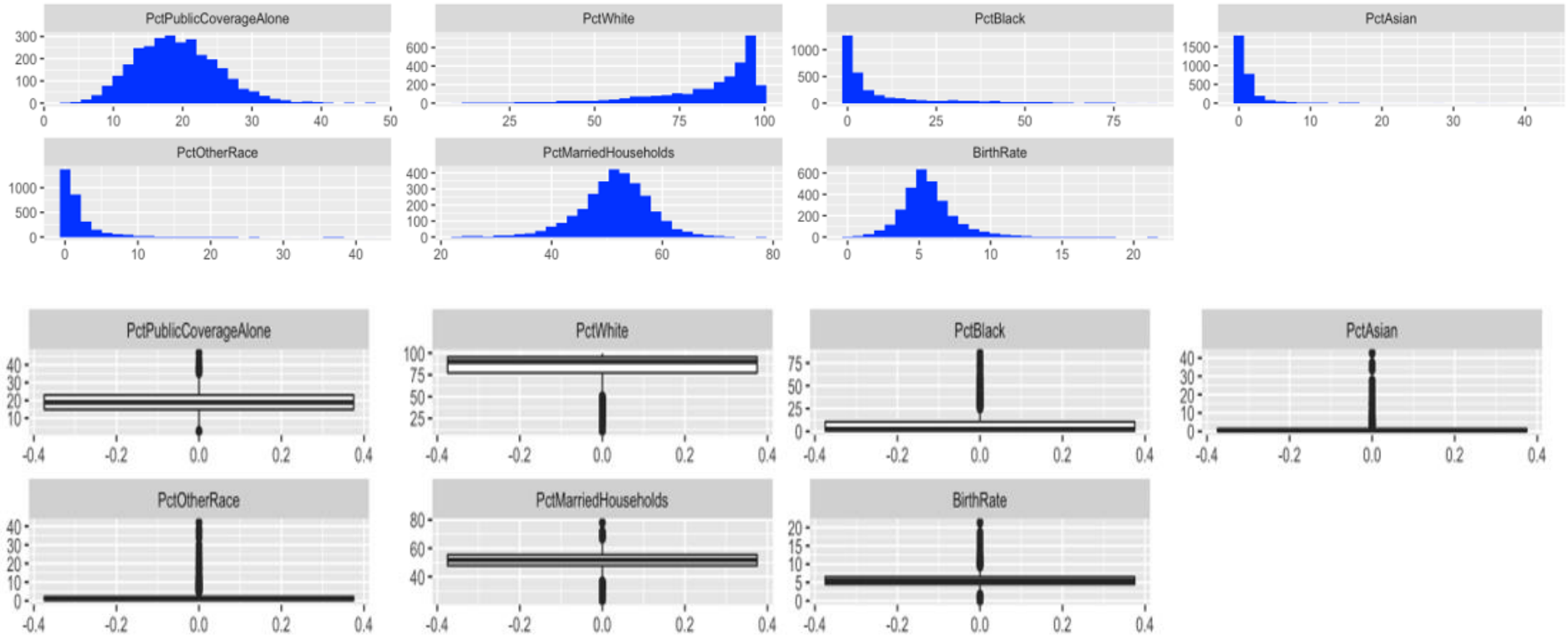


Figure 7

From the univariate graphs above, we can infer the following:

- 19.17 percent of county residents are with government-provided health coverage alone. (PctPublicCoverageAlone)
- The median percent of county residents who identify as white are 90. (PctWhite)
- The median percent of county residents who identify as black are 2.31. (PctBlack)
- 0.54 percent of county residents identify as Asian. (PctAsian)
- 0.84 percent of county residents identify in a category which is not white, black, or Asian. (PctOtherRace)
- 51.39 percent are married households. (PctMarriedHouseholds)
- The mean number of live births relative to number of women in county is 5.6. (BirthRate)

DATA PROFILING

Figure 8 shows an overall summary statistic for all the numerical variables in the data set.

```
 avgAnnCount      avgDeathsPerYear  TARGET_deathRate incidenceRate     medIncome       popEst2015       povertyPercent   studyPerCap
Min.   :    6.0   Min.   :    3.0   Min.   :110.4    Min.   :201.3   Min.   : 22640   Min.   :     827   Min.   : 3.20   Min.   :   0.00
1st Qu.:   78.0   1st Qu.:   29.0   1st Qu.:161.7    1st Qu.:421.8   1st Qu.: 39175   1st Qu.:   12162   1st Qu.:12.10   1st Qu.:   0.00
Median :  176.0   Median :   63.0   Median :178.1    Median :453.5   Median : 45430   Median :   27306   Median :15.80   Median :   0.00
Mean   :  621.4   Mean   :  191.2   Mean   :178.3    Mean   :448.2   Mean   : 47307   Mean   :  105489   Mean   :16.66   Mean   : 157.88
3rd Qu.:  527.8   3rd Qu.:  152.0   3rd Qu.:194.5    3rd Qu.:480.3   3rd Qu.: 52621   3rd Qu.:   70266   3rd Qu.:20.20   3rd Qu.:  86.06
Max.   :38150.0   Max.   :14010.0   Max.   :245.2    Max.   :718.9   Max.   :125635   Max.   :10170292   Max.   :46.90   Max.   :9762.31
  MedianAge     MedianAgeMale   MedianAgeFemale AvgHouseholdSize PercentMarried   PctNoHS18_24     PctHS18_24    PctBachDeg18_24   PctHS25_Over
Min.   :28.30   Min.   :23.70   Min.   :28.20   Min.   :0.0221   Min.   :23.10   Min.   : 0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 7.50
1st Qu.:37.80   1st Qu.:36.40   1st Qu.:39.10   1st Qu.:2.3700   1st Qu.:47.90   1st Qu.:12.80   1st Qu.:29.30   1st Qu.: 3.100   1st Qu.:30.60
Median :40.90   Median :39.50   Median :42.40   Median :2.5000   Median :52.50   Median :17.20   Median :34.70   Median : 5.400   Median :35.40
Mean   :40.84   Mean   :39.56   Mean   :42.18   Mean   :2.4802   Mean   :51.92   Mean   :18.25   Mean   :35.05   Mean   : 6.163   Mean   :34.95
3rd Qu.:43.80   3rd Qu.:42.40   3rd Qu.:45.30   3rd Qu.:2.6300   3rd Qu.:56.40   3rd Qu.:22.60   3rd Qu.:40.58   3rd Qu.: 8.200   3rd Qu.:39.70
Max.   :53.40   Max.   :53.40   Max.   :59.60   Max.   :3.9700   Max.   :72.50   Max.   :64.10   Max.   :72.50   Max.   :51.800   Max.   :54.80
PctBachDeg25_Over PctEmployed16_Over PctUnemployed16_Over PctPrivateCoverage PctPrivateCoverageAlone PctEmpPrivCoverage PctPublicCoverage
Min.   : 2.70     Min.   :17.6       Min.   : 0.400       Min.   :23.40      Min.   :16.80           Min.   :14.30      Min.   :11.20
1st Qu.: 9.40     1st Qu.:49.3       1st Qu.: 5.500       1st Qu.:57.50      1st Qu.:43.30           1st Qu.:34.83      1st Qu.:31.00
Median :12.30     Median :54.2       Median : 7.500       Median :65.15      Median :48.45           Median :41.40      Median :36.30
Mean   :13.22     Mean   :54.4       Mean   : 7.751       Mean   :64.51      Mean   :48.57           Mean   :41.48      Mean   :36.15
3rd Qu.:16.00     3rd Qu.:60.1       3rd Qu.: 9.600       3rd Qu.:72.10      3rd Qu.:53.80           3rd Qu.:47.70      3rd Qu.:41.30
Max.   :37.80     Max.   :80.1       Max.   :29.400       Max.   :92.30      Max.   :78.90           Max.   :70.70      Max.   :62.70
PctPublicCoverageAlone    PctWhite         PctBlack          PctAsian        PctOtherRace     PctMarriedHouseholds   BirthRate
Min.   : 2.60          Min.   : 11.01   Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   :22.99        Min.   : 0.000
1st Qu.:14.90          1st Qu.: 77.53   1st Qu.: 0.6424   1st Qu.: 0.2574   1st Qu.: 0.2981   1st Qu.:47.93        1st Qu.: 4.547
Median :18.80          Median : 90.00   Median : 2.3122   Median : 0.5473   Median : 0.8438   Median :51.74        Median : 5.394
Mean   :19.17          Mean   : 83.77   Mean   : 9.0987   Mean   : 1.2541   Mean   : 2.0044   Mean   :51.39        Mean   : 5.640
3rd Qu.:23.00          3rd Qu.: 95.37   3rd Qu.:10.5378   3rd Qu.: 1.1982   3rd Qu.: 2.1957   3rd Qu.:55.43        3rd Qu.: 6.473
Max.   :46.60          Max.   :100.00   Max.   :85.9478   Max.   :42.6194   Max.   :41.9303   Max.   :78.08        Max.   :21.326
```

Figure 8

BIVARIATE AND MULTIVARIATE ANALYSIS

A correlation matrix/heat map of the data set as shown in Figure 9 represents the pearson correlation coefficient value ranging between -1 and 1 that indicates the extent to which two variables are linearly related where -1 indicates perfect negative linear relationship, correlation of 0 indicates that two variables don't have any linear relationship whatsoever and correlation coefficient of 1 mean two variables are perfectly positively linearly related.

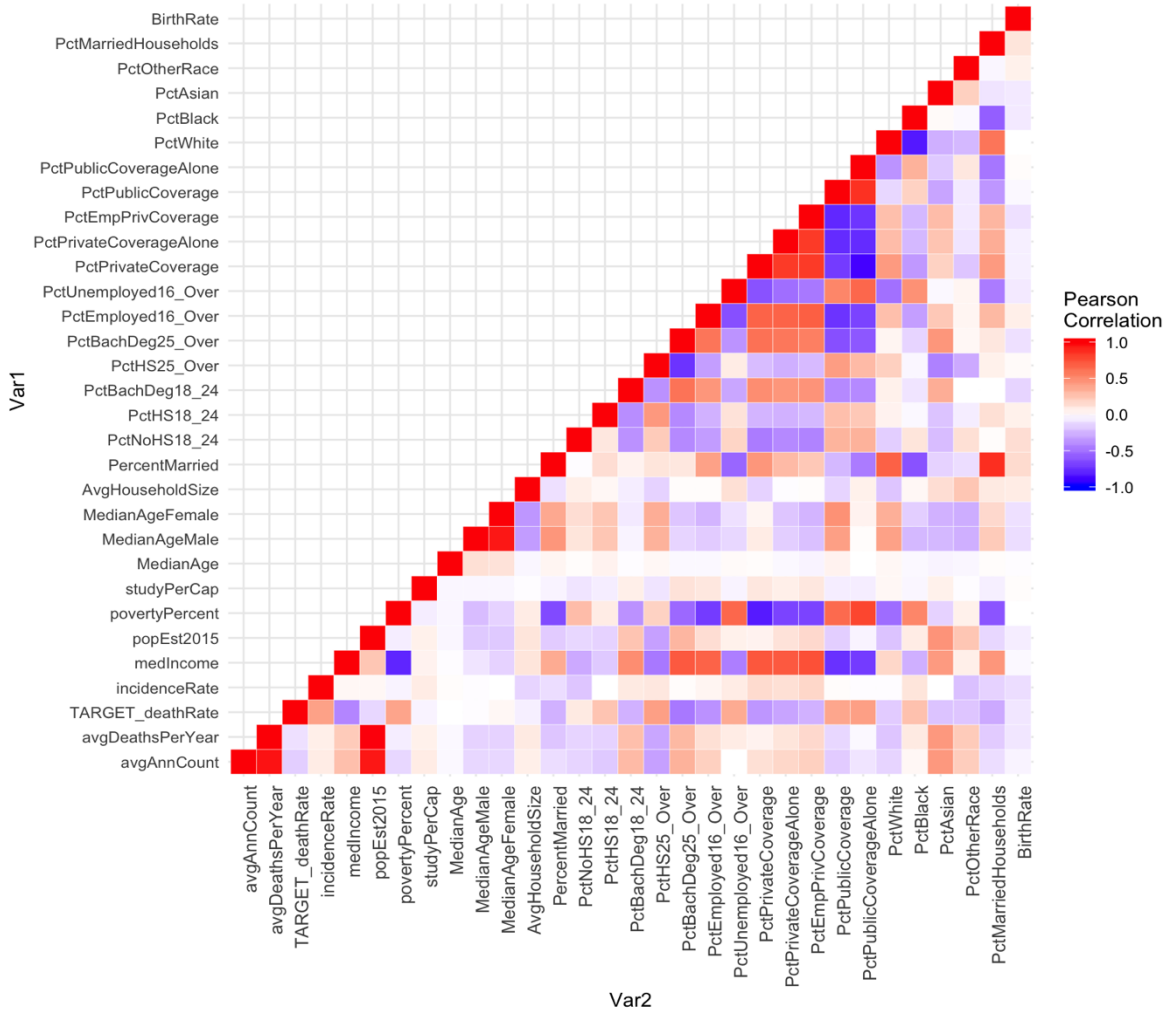
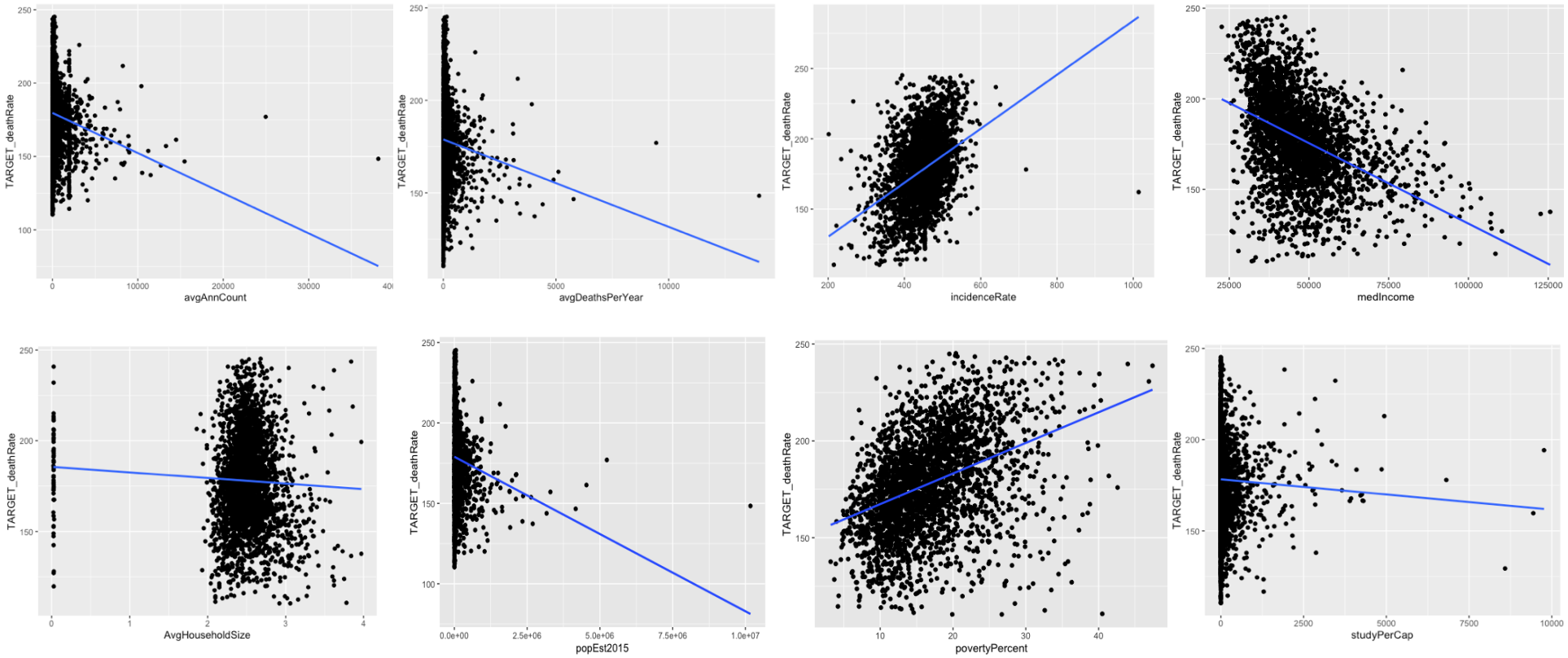
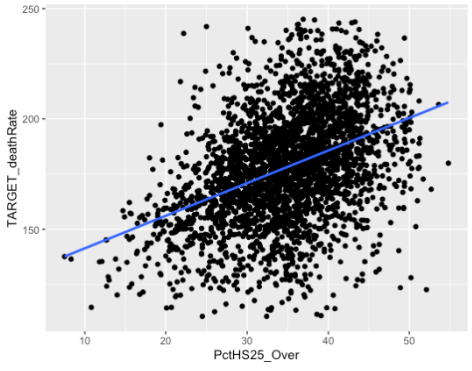


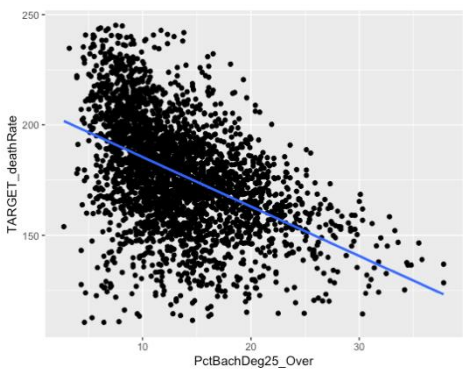
Figure 9

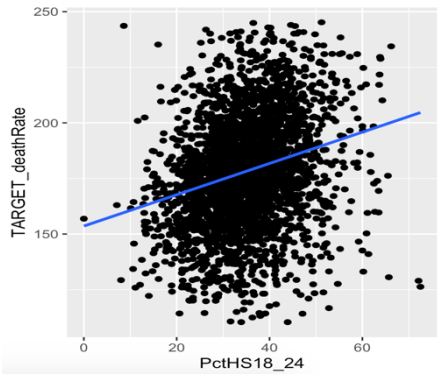
We see moderate to strong correlations for TARGET_deathRate with incidenceRate, povertyPercent, PctPublicCoverageAlone and strong correlation for PctPublicCoverage with PctPublicCoverageAlone and PctEmpPrivCoverage, PercentMarried with PctMarriedHouseholds, avgDeathsPerYear with avgAnnCount and popEst2015.

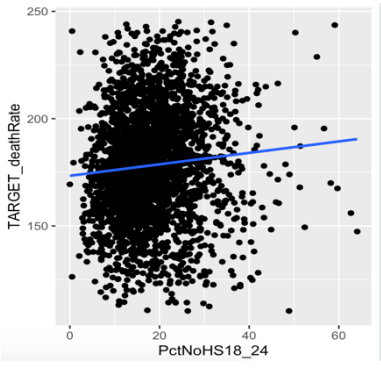
These relationships can also be observed visually with the scatterplot matrix as shown in Figure 10. It consists of a collection of scatterplots for each variable-combination of cancer dataset. Each of the scatterplot in the matrix pictures the relationship between a pair of variables which allows many relationships to be explored in just one diagram. If the data points make a straight line going from the left corner out to high x- and y-values, then the variables are said to have a positive correlation**.** If the data points go from a high-value on the y-axis down to a high-value on the x-axis, the variables have a negative correlation.

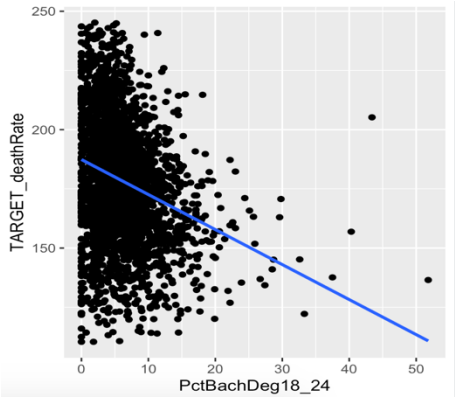


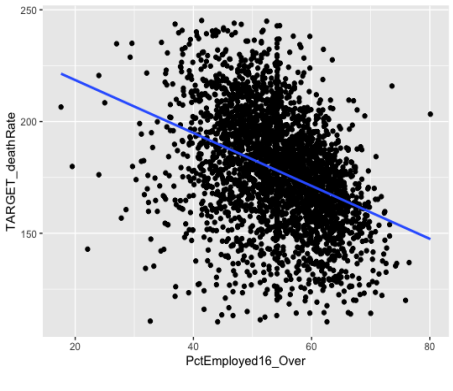

TARGET_deathRate
PctHS25_Over

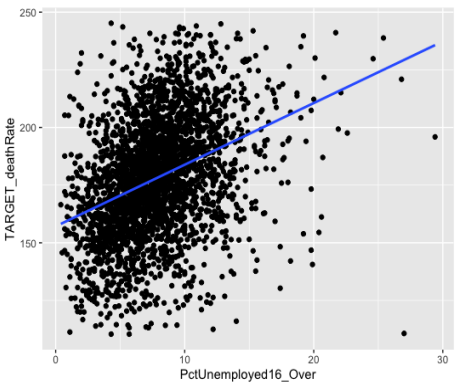

TARGET_deathRate
PctBachDeg25_Over

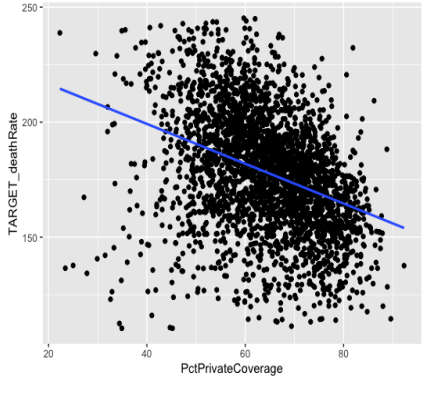

TARGET_deathRate
PctHS18_24

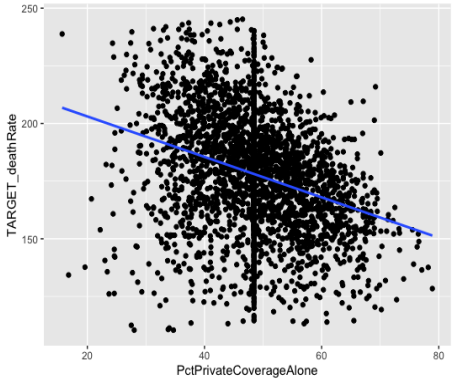

TARGET_deathRate
PctNoHS18_24

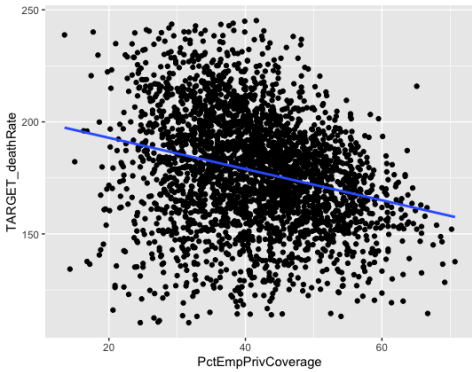

TARGET_deathRate
PctBachDeg18_24

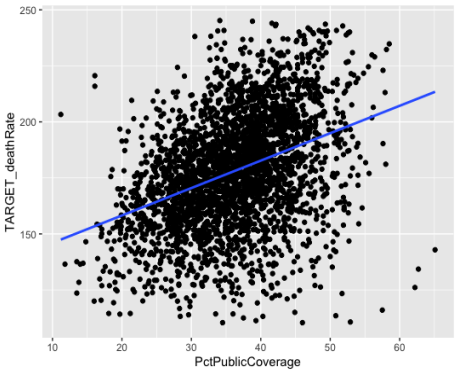

TARGET_deathRate
PctEmployed16_Over

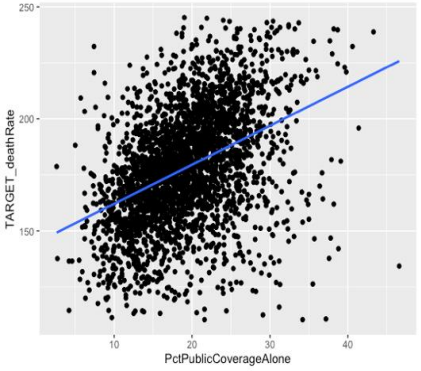

TARGET_deathRate
PctUnemployed16_Over


TARGET_deathRate
PctPrivateCoverage


TARGET_deathRate
PctPrivateCoverageAlone


TARGET_deathRate
PctEmpPrivCoverage


TARGET_deathRate
PctPublicCoverage


TARGET_deathRate
PctPublicCoverageAlone

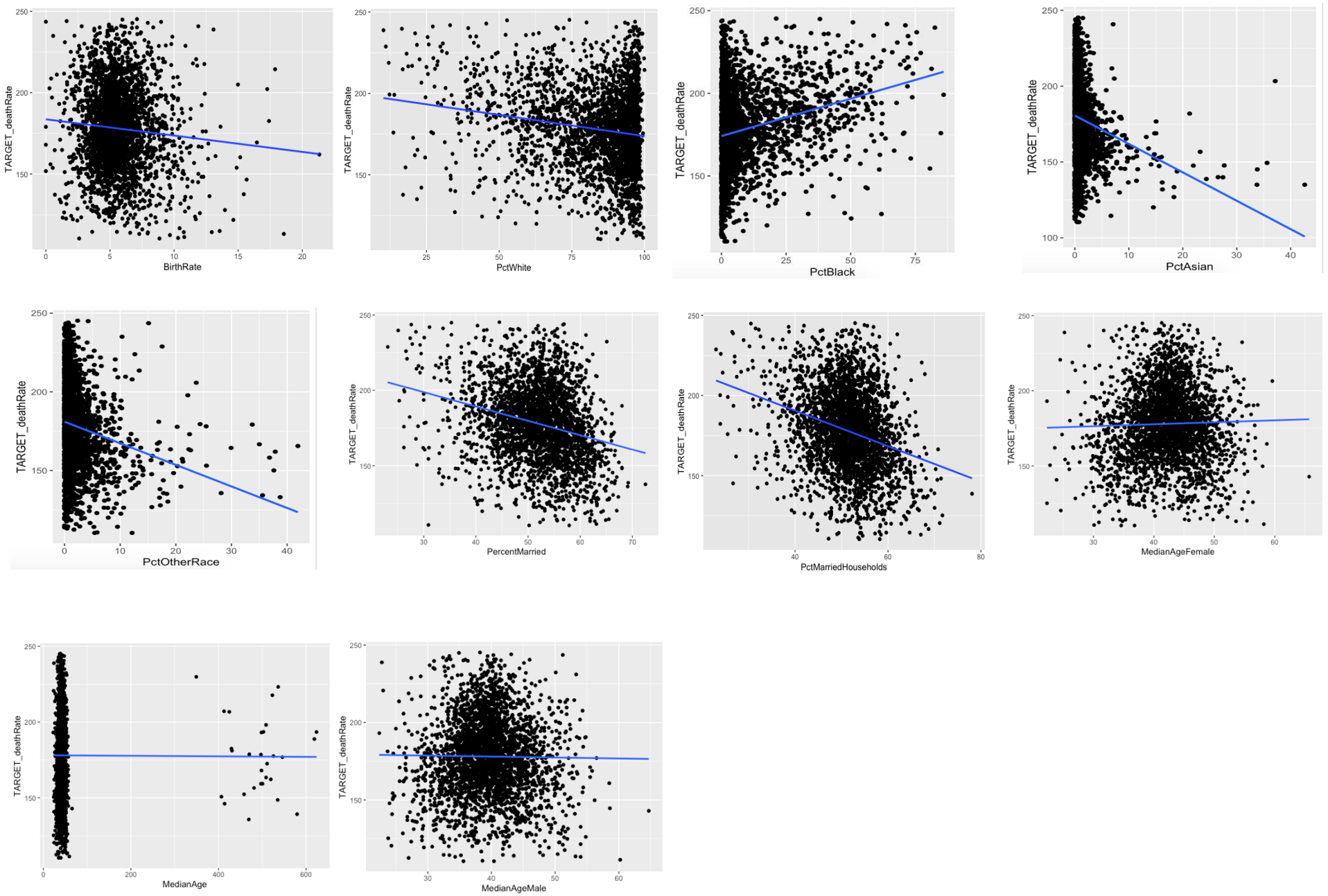

TARGET_deathRate
BirthRate
PctWhite
PctBlack
PctAsian
PctOtherRace
PercentMarried
PctMarriedHouseholds
MedianAgeFemale
MedianAge
MedianAgeMale

Figure 10

The following can be inferred from the scatterplots of all numeric variables with target variable death rate (Figure 10):

1. Target death rate appears to fall in a curvilinear manner with increase in median income.
2. Target death rate increases (slightly) linearly with povertyPercent.
3. There does not seem to be conclusive relationship between population of county and target death rate.
4. No concluding pattern between death rate and per capita number of cancer-related clinical trials.
5. Target death rate does not have any conclusive relationship with MedianAgeMale, MedianAgeFemale and MedianAge.
6. There does not seem to be a clear relationship between AvgHouseholdSize and death rate.
7. None of the three 18-24 age group variables show any distinct pattern with respect to death rate.
8. There seems to be slight increase in death rate with higher percentage of 25 and over high school graduate.
9. There is a curvilinear relationship(decreasing) between death rate and percentage over 25 with bachelor's degree.
10. Death rate decreases slightly with increase in PctEmployed16_Over and increases slightly with increase in PctUnemployed16_Over.
11. There appears to be weak linear relationship with private coverage variables and death rate and there exists a linear relationship between death rate and public coverage.
12. People with PctPrivateCoverageAlone seem to have a lower death rate.
13. There seems to no trend visible between death rate and any of the race except for minimal increase in death rate with PctBlack.
14. There appears to be a very slight decrease in death rate with PercentMarried and also Plotting PctMarriedHouseholds against deathrate shows marginal decrease with increase in PctMarriedHousehold.
15. Birthrate of county shows no apparent relationship to cancer death rate.

Let’s look at some of the multivariate graphs. The relationship between state and median income per state with cancer death rate intimate that residents with a higher median income of the state has lower cancer death rates and people with a low median income of the state have high cancer death rates.

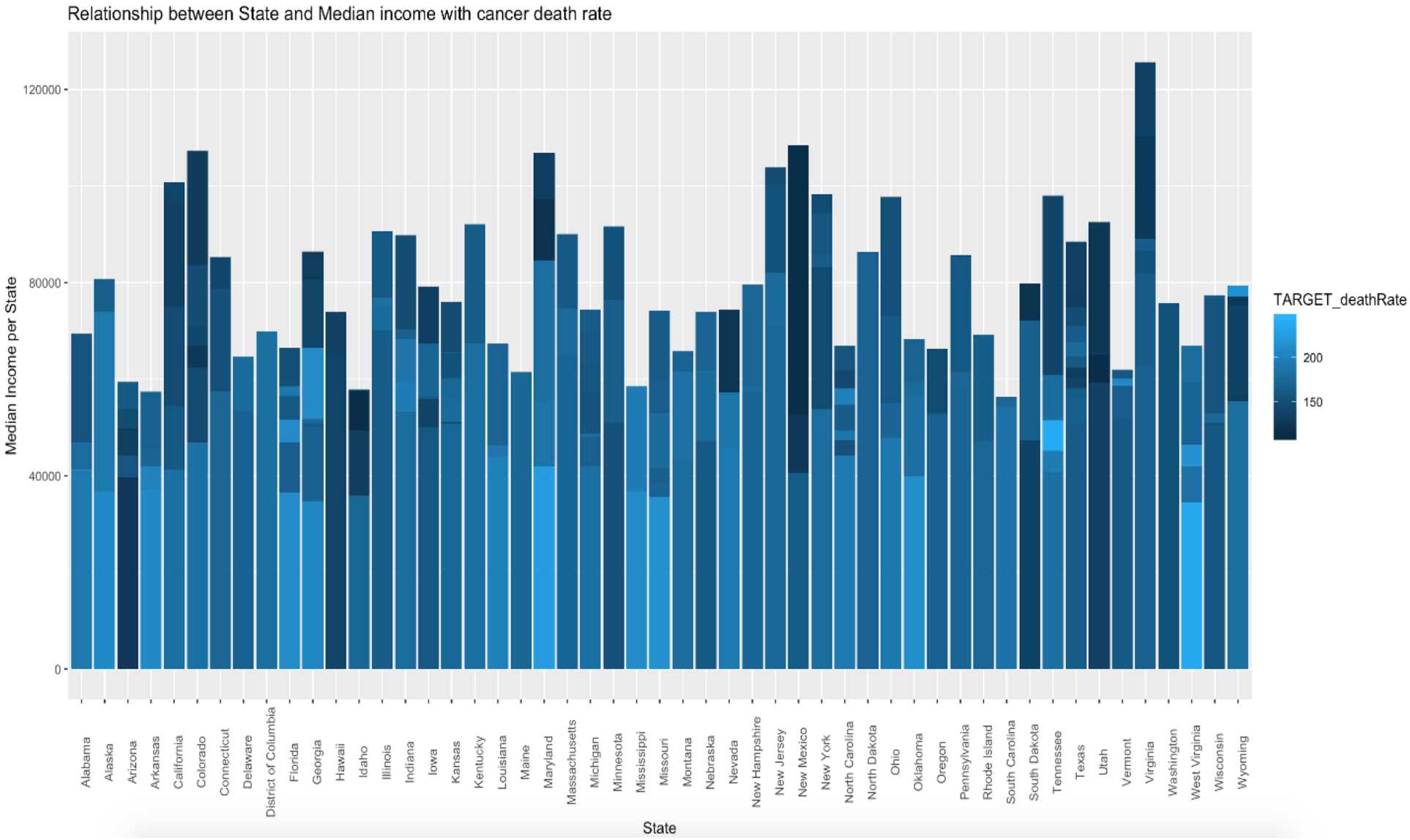


Figure 11

The relationship between state and residents with private coverage alongside cancer death rate intimate that counties with residents having high private coverage percentage show a low cancer death rate.

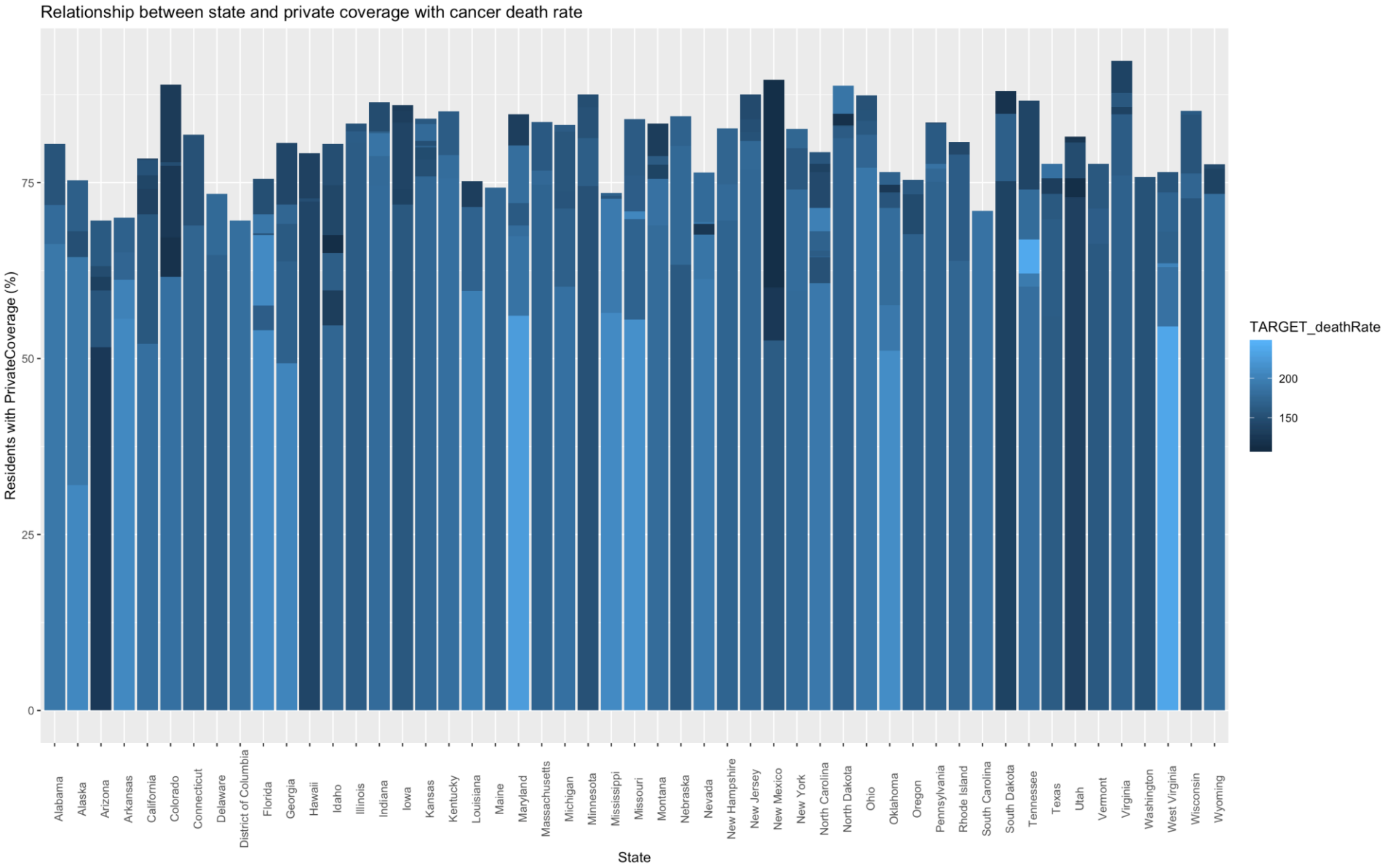


Figure 12

The relationship between state and residents with public coverage only and no private assistance alongside cancer death rate intimate that counties with residents having high public coverage only show a high cancer death rate.

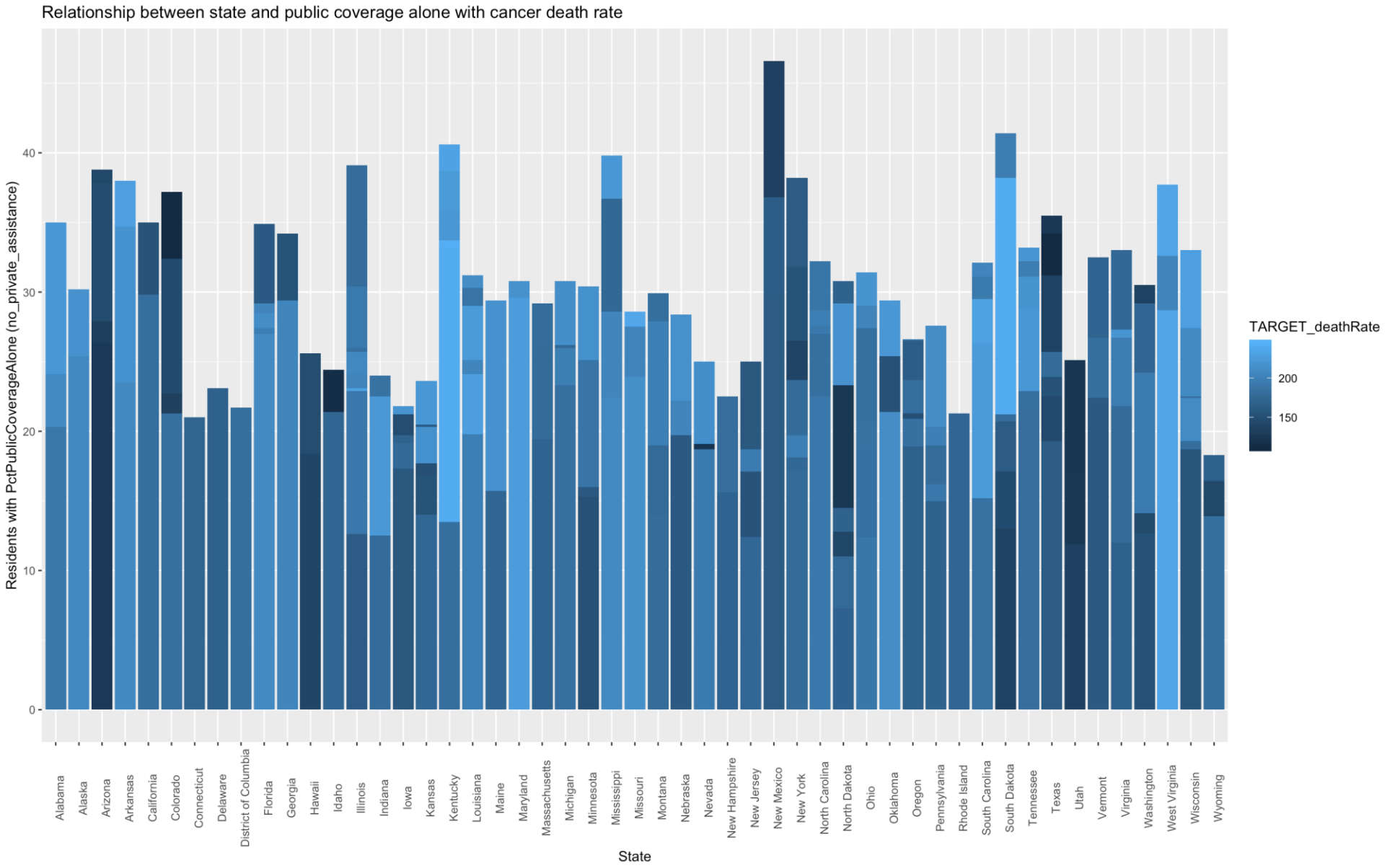


Figure 13

Cancer death rates by US state map

We have also looked at the cancer mortality rates by US state map per 100,000 people. The color coding used here is white to red which means that shades of white show low death rates and shade of red indicate high death rates.

From Figure 15, we can sight that the highest cancer death rates among the US states is Texas (TX) color coded by red and the next is Georgia (GA) color coded by lighter shade of red. The least cancer mortality rates are evident in some of the states such as Delaware (DE), Hawaii (HI), Connecticut (CT) followed by Arizona (AZ) and many more.

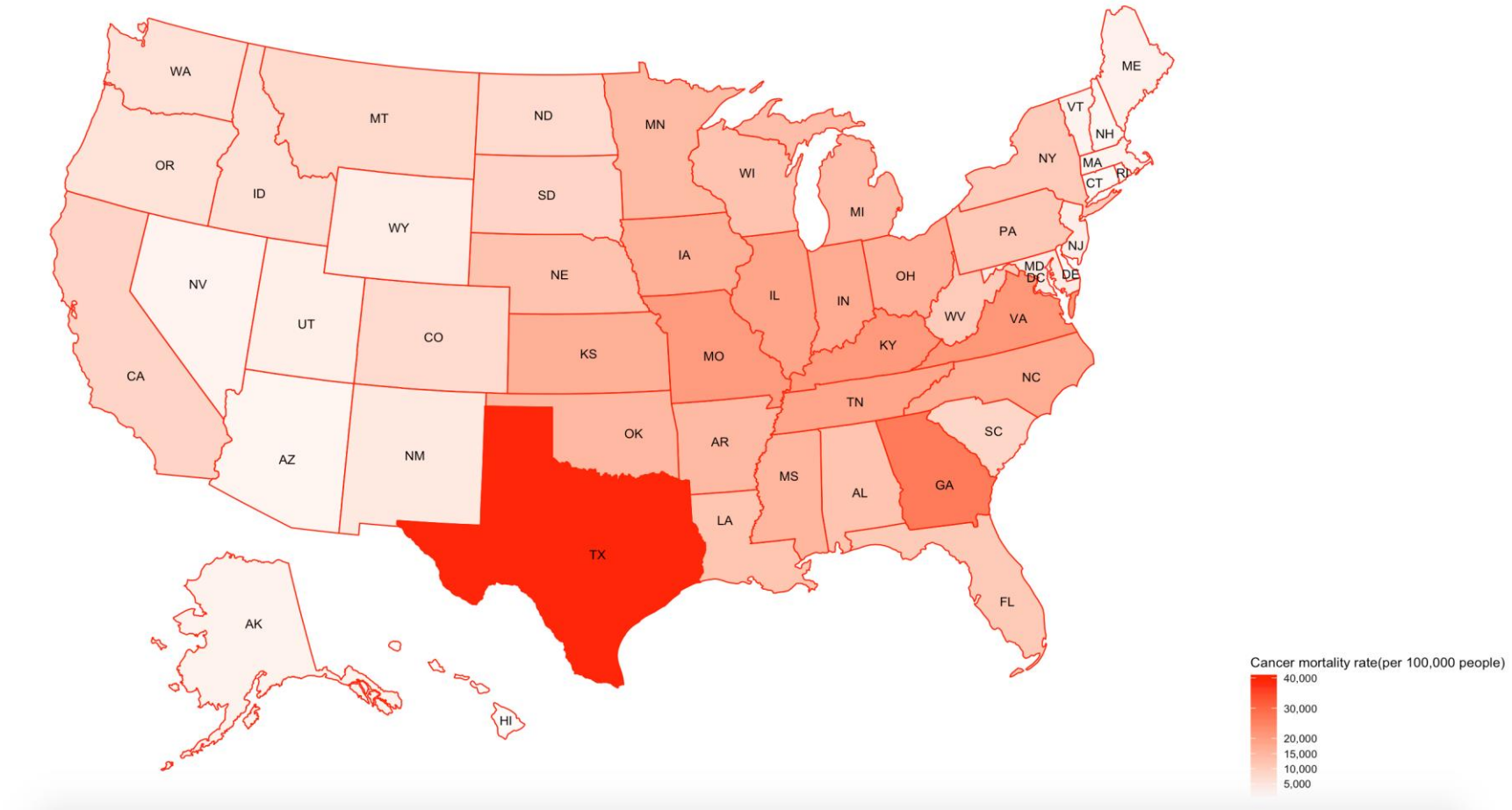
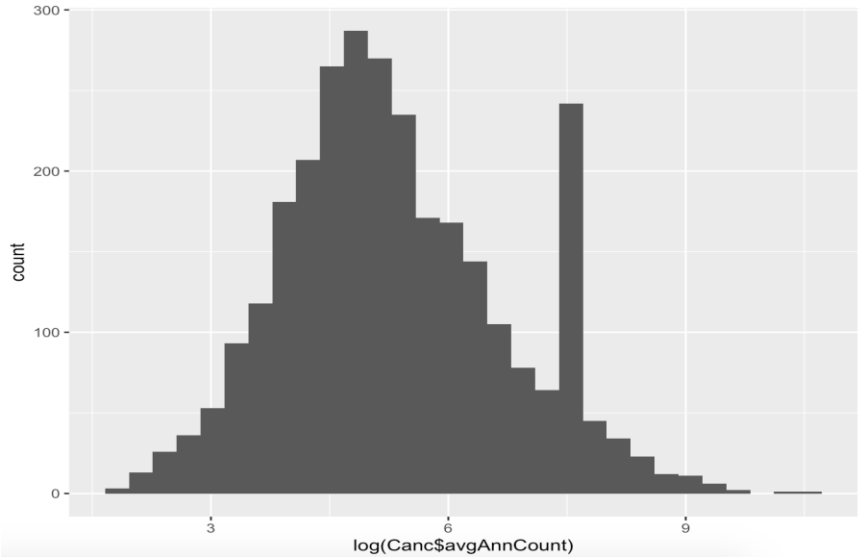
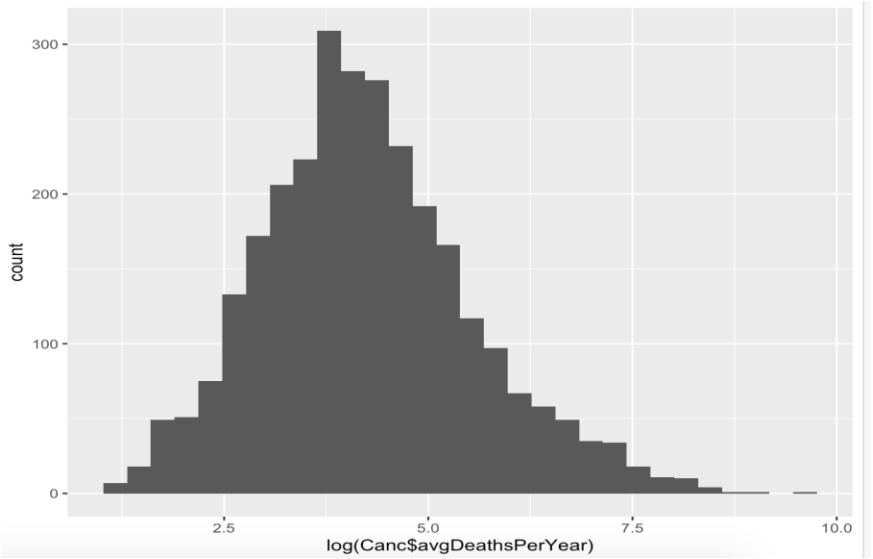


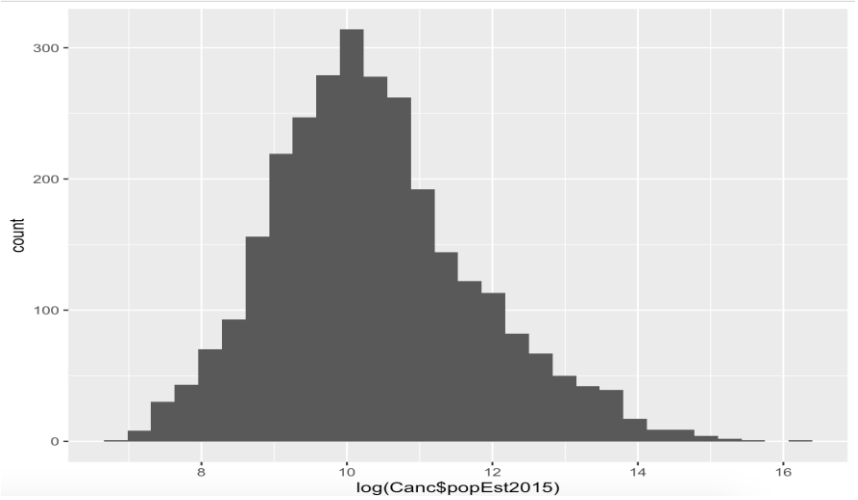
Figure 15

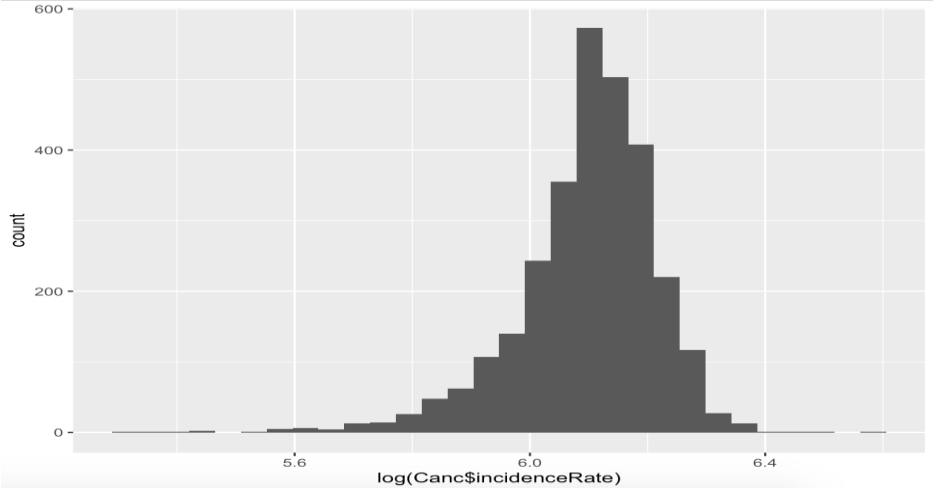
DATA TRANSFORMATION

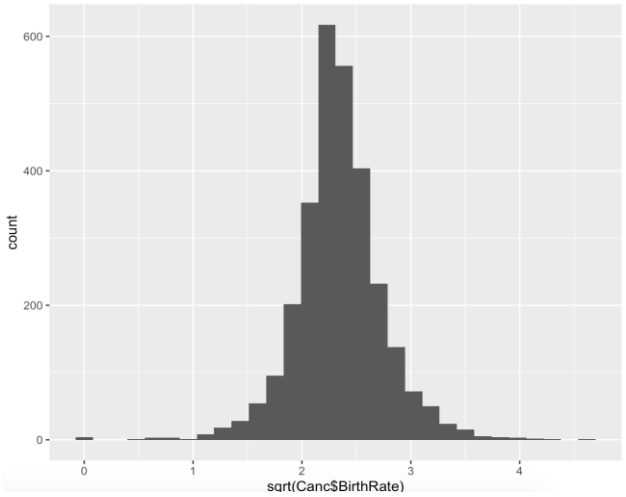
Relevant transformation is applied on the variables that were highly skewed. The transformation helps to change a highly skewed variable into a more normalized data.
Figure 16 shows the respective transformation that were applied.

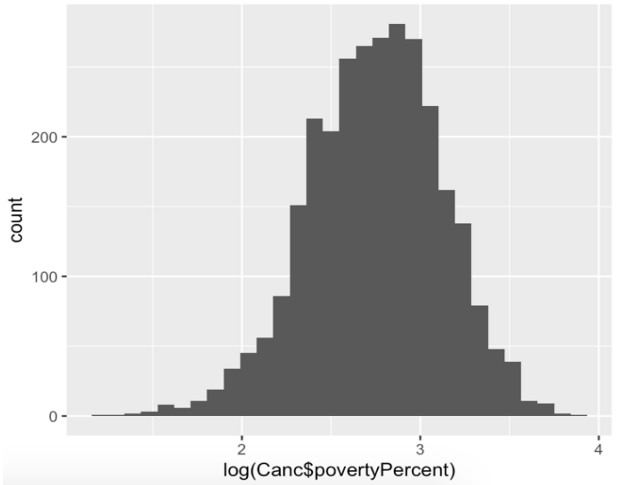


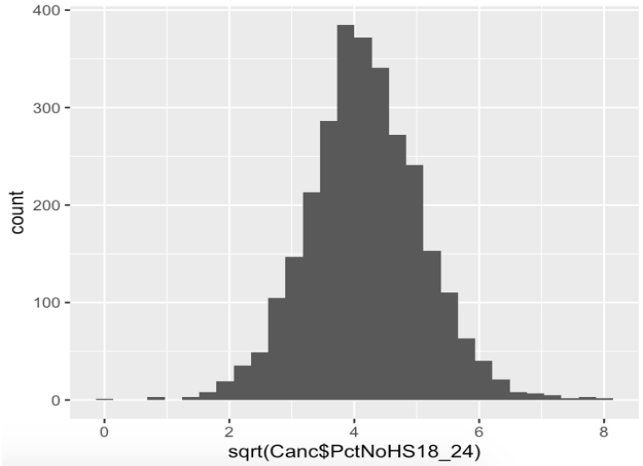


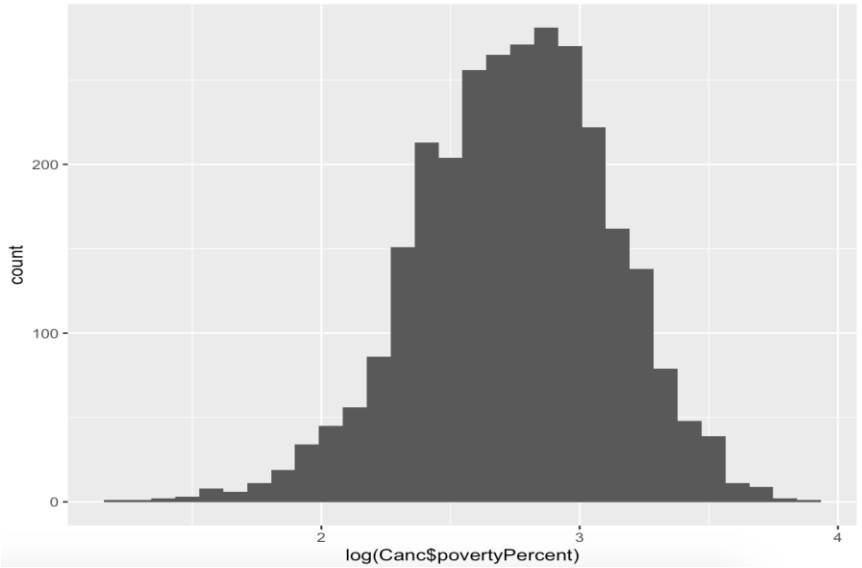





count
sqrt(Canc$BirthRate)


count
log(Canc$povertyPercent)


count
sqrt(Canc$PctNoHS18_24)


count
log(Canc$povertyPercent)

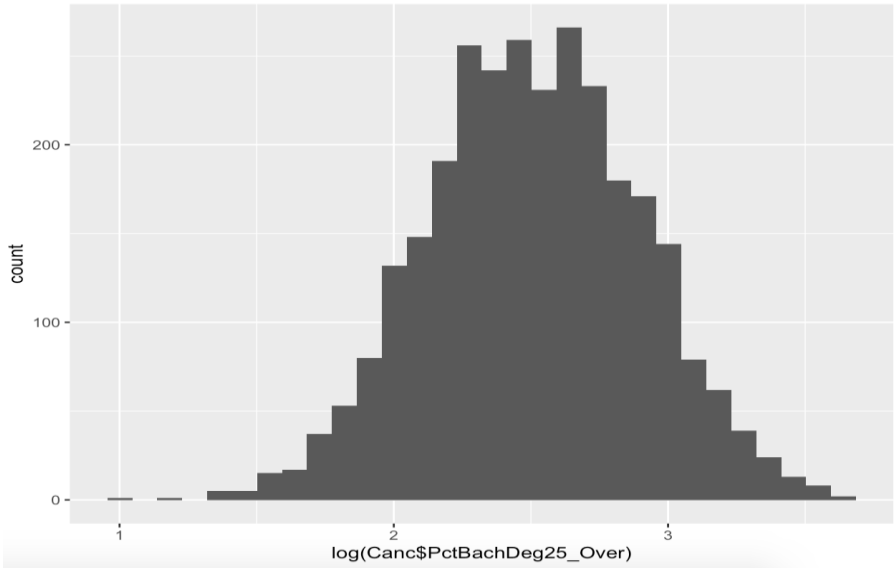

count
log(Canc$PctBachDeg25_Over)

Figure 16

MODEL BUILDING

Our goal is to construct a model to predict Cancer Mortality Rates in US based on available data which is socio-economic factors. We will build a model to predict the mortality rate based on the given variables. Since we have multiple variables to predict the target death rate value, we will be using multiple linear regression which is one of the core methods for developing a model.

MODEL DIAGNOSIS AND SELECTION

For the model 1 we used stepwise regression approach to find the best model. For this model, we started by using all the variables except categorical variables because the model with categorical variable showed very high VIF for all the categorical columns

```
summary(step.model)

Call:
lm(formula = Canc$TARGET_deathRate ~ avgAnnCount + avgDeathsPerYear +
    incidenceRate + popEst2015 + MedianAgeMale + PercentMarried +
    PctNoHS18_24 + PctHS18_24 + PctHS25_Over + PctBachDeg25_Over +
    PctEmployed16_Over + PctUnemployed16_Over + PctPrivateCoverage +
    PctEmpPrivCoverage + PctWhite + PctOtherRace + PctMarriedHouseholds +
    BirthRate, data = Canc)

Residuals:
    Min      1Q  Median      3Q     Max
-86.889 -10.375  -0.205  10.710  81.929

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.806e+02  8.035e+00  22.482  < 2e-16 ***
avgAnnCount        -4.003e-03  7.261e-04  -5.513 3.82e-08 ***
avgDeathsPerYear    1.987e-02  3.604e-03   5.512 3.86e-08 ***
incidenceRate       1.672e-01  7.094e-03  23.565  < 2e-16 ***
popEst2015         -1.534e-05  5.002e-06  -3.067  0.00219 **
MedianAgeMale      -7.343e-01  1.025e-01  -7.164 9.86e-13 ***
PercentMarried      1.034e+00  1.553e-01   6.657 3.31e-11 ***
PctNoHS18_24       -9.339e-02  5.090e-02  -1.835  0.06663 .
PctHS18_24          1.896e-01  4.416e-02   4.294 1.81e-05 ***
PctHS25_Over        4.657e-01  8.915e-02   5.224 1.87e-07 ***
PctBachDeg25_Over  -1.031e+00  1.389e-01  -7.426 1.45e-13 ***
PctEmployed16_Over -4.213e-01  8.308e-02  -5.071 4.19e-07 ***
PctUnemployed16_Over  2.596e-01  1.536e-01   1.690  0.09109 .
PctPrivateCoverage -5.130e-01  8.170e-02  -6.280 3.89e-10 ***
PctEmpPrivCoverage  3.301e-01  8.264e-02   3.995 6.64e-05 ***
PctWhite           -1.002e-01  3.133e-02  -3.199  0.00139 **
PctOtherRace       -8.392e-01  1.109e-01  -7.569 4.98e-14 ***
PctMarriedHouseholds -1.061e+00  1.332e-01  -7.964 2.35e-15 ***
BirthRate          -7.713e-01  1.828e-01  -4.219 2.53e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 18.04 on 2964 degrees of freedom
Multiple R-squared:  0.484,    Adjusted R-squared:  0.4809
F-statistic: 154.5 on 18 and 2964 DF,  p-value: < 2.2e-16
```

```
> vif(step.model)
       avgAnnCount     avgDeathsPerYear        incidenceRate           popEst2015        MedianAgeMale       PercentMarried
          9.855756           30.832165            1.227944           25.311129            2.619867           10.285811
      PctNoHS18_24          PctHS18_24         PctHS25_Over    PctBachDeg25_Over   PctEmployed16_Over PctUnemployed16_Over
          1.532981            1.441816            3.554391            4.888902            3.986530            2.477119
 PctPrivateCoverage  PctEmpPrivCoverage          PctWhite         PctOtherRace PctMarriedHouseholds           BirthRate
          6.721729            5.533695            2.348188            1.404752            6.896936            1.181947
```

After conducting the summary and model analysis and checking VIF for the model 1, we found out some of the findings which are as follows:

- The p-values for all variables except PctNoHS18_24,PctUnemployed16_Over is more than 0.05, which means that at the 95% significance level, those variables are not significant.

- TheVIF values for avgAnnCount, avgDeathsPerYear, popEst2015, PercentMarried, PctPrivateCoverage, PctEmpPrivCoverage, PctMarriedHouseholds are more than the selected threshold, 5. This means we cannot use these variables in the same model since it leads to a problem of multi-collinearity.

- Summary also provide R-squared and Adjusted R-squared value. R-squared indicates the proportion of the variation in your dependent variable (Y) explained by your independent variables (X) for a linear regression model which can be interpreted as 0.4809 or 48.09 %.R-square will be increased simply by adding additional predictors to the model, thus adjusted R-Squared is used instead of R-squared for comparing models with more than one predictor variable.

MODEL 2

```
> summary(step.model)

Call:
lm(formula = Canc2$TARGET_deathRate ~ avgAnnCount + avgDeathsPerYear +
    incidenceRate + popEst2015 + MedianAgeMale + PercentMarried +
    PctHS18_24 + PctHS25_Over + PctBachDeg25_Over + PctPrivateCoverage +
    PctEmpPrivCoverage + PctOtherRace + PctMarriedHouseholds +
    BirthRate, data = Canc2)

Residuals:
    Min      1Q  Median      3Q     Max
-65.871  -8.937  -0.743   7.895 150.903

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          7.518e+02  1.400e+01  53.720  < 2e-16 ***
avgAnnCount         -3.168e+00  3.264e-01  -9.707  < 2e-16 ***
avgDeathsPerYear     8.232e+01  1.767e+00  46.588  < 2e-16 ***
incidenceRate        9.797e-02  5.796e-03  16.903  < 2e-16 ***
popEst2015          -7.792e+01  1.759e+00 -44.291  < 2e-16 ***
MedianAgeMale       -2.595e+00  8.461e-02 -30.670  < 2e-16 ***
PercentMarried      -5.134e-03  9.759e-04  -5.261 1.53e-07 ***
PctHS18_24           3.774e-01  3.363e-02  11.220  < 2e-16 ***
PctHS25_Over        -3.151e-01  6.699e-02  -4.704 2.67e-06 ***
PctBachDeg25_Over   -7.255e+00  1.342e+00  -5.408 6.88e-08 ***
PctPrivateCoverage  -6.554e-01  6.189e-02 -10.590  < 2e-16 ***
PctEmpPrivCoverage   5.624e-01  6.235e-02   9.020  < 2e-16 ***
PctOtherRace        -3.063e+00  4.632e-01  -6.612 4.47e-11 ***
PctMarriedHouseholds 6.047e-03  9.244e-04   6.542 7.12e-11 ***
BirthRate           -1.921e+00  6.675e-01  -2.878  0.00403 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.92 on 2968 degrees of freedom
Multiple R-squared:  0.6923,   Adjusted R-squared:  0.6909
F-statistic: 477.1 on 14 and 2968 DF,  p-value: < 2.2e-16
```

```
> vif(step.model)
        avgAnnCount     avgDeathsPerYear        incidenceRate          popEst2015       MedianAgeMale      PercentMarried
           3.321669           81.665443            1.376696           94.038452            2.997637            6.802124
         PctHS18_24        PctHS25_Over   PctBachDeg25_Over  PctPrivateCoverage PctEmpPrivCoverage        PctOtherRace
           1.404179            3.370395            4.175691            6.478321            5.289708            1.534251
PctMarriedHouseholds           BirthRate
           5.577580            1.159588
>
```

After conducting the summary and model analysis and checking VIF for the model 1, we found out some of the findings which are as follows:

- The p-values for all the variables where significant because each variable is less than 0.05.

- TheVIF values for avgDeathsPerYear, popEst2015, PercentMarried, PctPrivateCoverage, are more than the selected threshold, 5. This means we cannot use these variables in the same model since it leads to a problem of multi-collinearity.

- Summary also provide R-squared and Adjusted R-squared value. R-squared indicates the proportion of the variation in your dependent variable (Y) explained by your independent variables (X) for a linear regression model which can be interpreted as 0.6909 or 69.09 %.R-square will be increased simply by adding additional predictors to the model, thus adjusted R-Squared is used instead of R-squared for comparing models with more than one predictor variable.

FEATURE ENGINEERING:

To handle outliers and missing values using GLRM

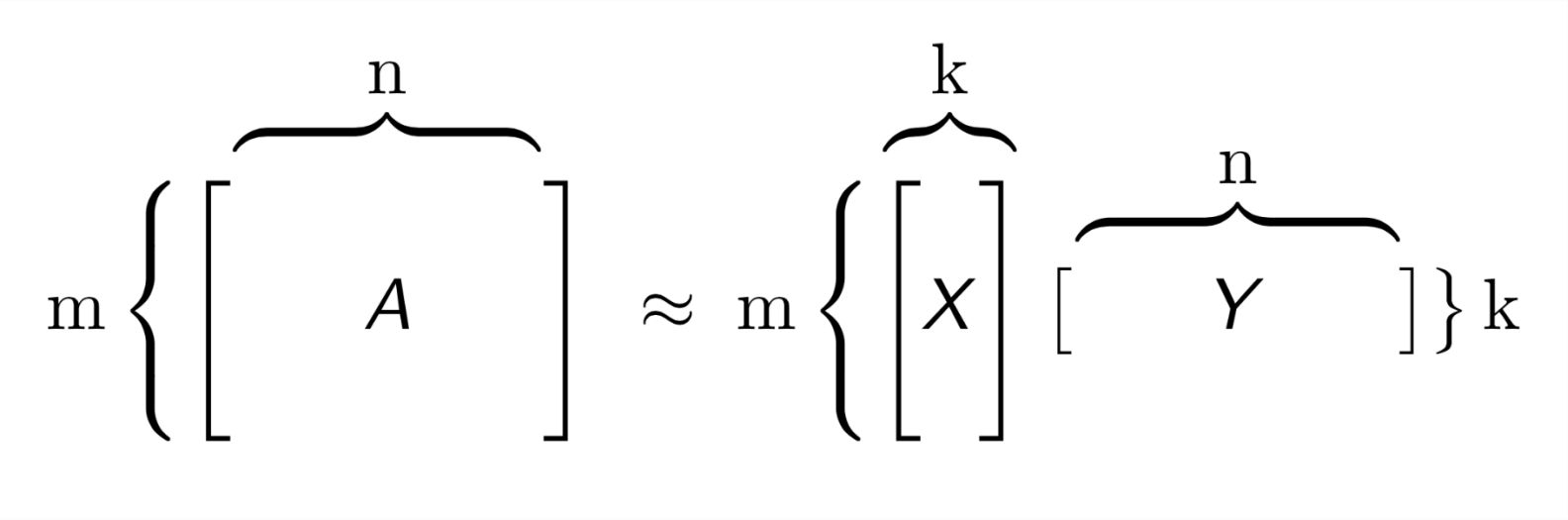
Generalized Low Rank Models (GLRM) is an algorithm for dimensionality reduction of a dataset. It is a general, parallelized optimization algorithm that applies to a variety of loss and regularization functions. Categorical columns are handled by expansion into 0/1 indicator columns for each level. With this approach, GLRM is useful for reconstructing missing values and identifying important features in heterogeneous data.

```
# data cleaning
canc2$binnedInc<-as.factor(canc2$binnedInc)
canc2<- tidyr::separate(canc2,"Geography",into = c("County/City","State"),sep = ",")

# Impute missing values and check for outliers
h2o.init(nthreads = -1,min_mem_size = "4g")
canc2<- as.h2o(canc2)

canc.glrm<-h2o.glrm(training_frame = canc2,k=10,init = "SVD",svd_method = "GramSVD",max_iterations =3000,min_step_size = 1e-6 )
h2o.performance(canc.glrm)
canc.pred<-predict(canc.glrm,canc2)
canc.pred
feat.name<- setdiff(names(canc.pred),"reconstr_TARGET_deathRate")
feat.name
canc.dl<-h2o.deeplearning(x =feat.name, training_frame = canc.pred, autoencoder = T,reproducible = T,seed = 323,hidden = c(10,2,10),epochs = 100)
summary(canc.dl)
canc.anom<-h2o.anomaly(canc.dl,canc.pred)
canc.anom<-canc.anom %>% as.data.frame() %>% mutate(row_number=1:3047)%>% filter(Reconstruction.MSE>.09)

canc.pred<-h2o.cbind(canc.pred,canc2[,c("County/City","State")])
canc2<- canc.pred%>% as.data.table()
canc2$State<-as.factor(canc2$State)
canc2$County.City<-as.factor(canc2$County.City)

canc.anom# possible outliers

canc2[canc.anom$row_number]

canc2[new_MedianAge>100,new_MedianAge]#outliers

h2o.shutdown(F); rm(canc.glrm,canc.pred,canc.dl); gc();detach("package:h2o", unload=TRUE)
```

What is a Low-Rank Model?

Given large collections of data with numeric and categorical values, entries in the table may be noisy or even missing altogether. Low rank models facilitate the understanding of tabular data by producing a condensed vector representation for every row and column in the dataset. Specifically, given a data table A with m rows and n columns, a GLRM consists of a decomposition of A into numeric matrices X

and Y. The matrix X has the same number of rows as A, but only a small, user-specified number of columns k. The matrix Y has k rows and d columns, where d is equal to the total dimension of the embedded features in A.



Both X and Y have practical interpretations. Each row of Y is an archetypal feature formed from the columns of A, and each row of X corresponds to a row of A projected into this reduced dimension feature space. We can approximately reconstruct A from the matrix product XY, which has rank k. The number k is chosen to be much less than both m and n.

After trying stepwise AIC method as shown earlier, we did not get a high value of Adjusted R2, hence we decided to try GLRM using h20.

We use the original dataset, Canc2 and input that as the data in h20. For GLRM, we are using k=10 and are using the svd method for validation. After we run the glrm model, we get two matrices, X and Y as shown above with the order of 3047*10 and 10*34. GLRM handles missing values. Next, we predict the values of our dataset based on the generated model, which is stored in canc.pred. Now, canc.pred has brought back the data by multiplying the X and Y matrices and will be used as our dataset going forward, rid of all missing values and outliers handled. We also perform outlier detection using Deep Learning, which is shown in the code above. So, from here on we will use these reconstructed variables for model selection.

FEATURE SELECTION

```
#Feature Selection

#using the variables found from STEPAIC model earlier, but using reconstructed variables now
imp_feat<-c("reconstr_avgAnnCount",
            "reconstr_PercentMarried",
            "reconstr_PctEmployed16_Over",
            "reconstr_PctMarriedHouseholds",
            "reconstr_incidenceRate",
            "reconstr_PctHS18_24",
            "reconstr_PctPrivateCoverage",
            "reconstr_BirthRate",
            "reconstr_popEst2015",
            "reconstr_PctHS25_Over",
            "reconstr_PctEmpPrivCoverage",
            "reconstr_MedianAgeMale",
            "reconstr_PctBachDeg25_Over",
            "reconstr_PctOtherRace")
```

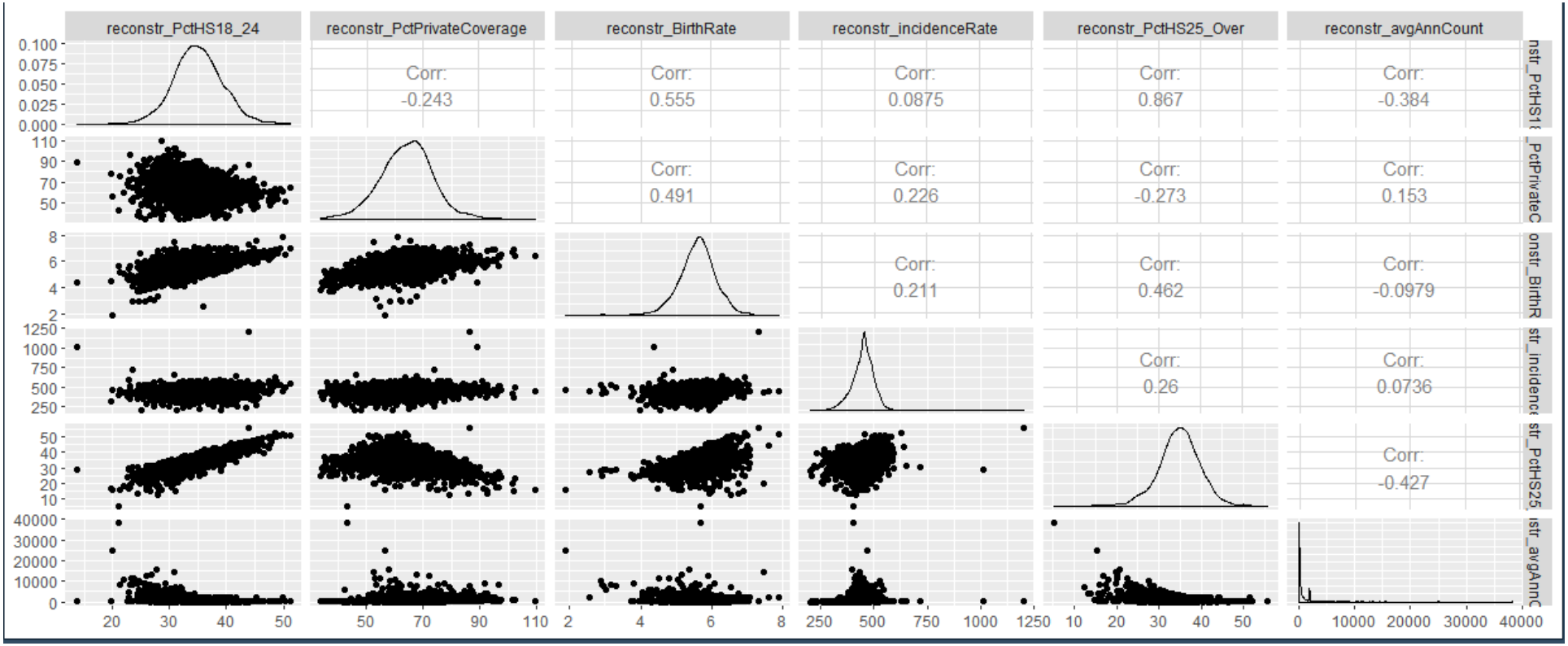
The reason we have chosen these 14 variables are because they were the variables that we got from the STEPAIC model we ran in the beginning. The reason we are re-doing the modeling again with the reconstructed variables are because the Adjusted R^2 was very low.

```
#Not all the variables are significant and VIF>5, so we remove some variables and try forward stepwise again
imp_feat<-c("reconstr_avgAnnCount",
            "reconstr_PercentMarried",
            "reconstr_PctEmployed16_Over",
            "reconstr_PctMarriedHouseholds",
            "reconstr_incidenceRate",
            "reconstr_PctHS18_24",
            "reconstr_PctPrivateCoverage",
            "reconstr_BirthRate",
            "reconstr_popEst2015",
            "reconstr_PctHS25_Over")
```

```
#finally reached these set of variables which had all high significance and VIF<5.

imp_feat<-c("reconstr_PctHS18_24",
            "reconstr_PctPrivateCoverage",
            "reconstr_BirthRate",
            "reconstr_incidenceRate",
            "reconstr_PctHS25_Over",
            "reconstr_avgAnnCount")
```

After performing forward stepwise on the first set of important features, we found that some variables are not significant and have very high VIF values. After plugging and removing, we reached the final set of variables that we will be using for the forward stepwise method modeling.

Correlation matrix for Feature selected:



From the correlation matrix, we can see that there is little to no correlation between the variables we will be using, except for the reconstr-PctHS25-Over and reconstr-PCTHS18-24.

Training- Test Split:

Here, we are creating a partition for splitting the data into training and testing. We are going to use 70% of the data as training and 30% of the data as testing set. Within this, we further split training set to x_train and y_train and split the test set into x_test and y_test.

```
intrain<-createDataPartition(canc2$reconstr_TARGET_deathRate,p=.70,list=F)
intrain
training<-canc2[intrain]
training
y_train=training$reconstr_TARGET_deathRate
y_train
x_train=training[,-3]
x_train

testing<-canc2[-intrain]
testing
y_test=testing$reconstr_TARGET_deathRate
y_test
x_test=testing[,-3]
x_test
```

The code below shows the execution of forward stepwise method for the important features we chose for modeling.

```
#this is essentially forward stepwise method
models<-list()
for(i in seq_along(imp_feat[-7])){


  models[[i]]<- lm(reconstr_TARGET_deathRate~.,data=training[,imp_feat[c(1:i,7)],with=F])


}
lapply(models,summary)

do.call(anova,models)
```

The below R code shows the output when we run the lapply function. This shows the 6 models we achieve using the for loop using Forward Stepwise.

```
[[1]]

Call:
lm(formula = reconstr_TARGET_deathRate ~ ., data = training[,
    imp_feat[c(1:i, 7)], with = F])

Residuals:
    Min      1Q  Median      3Q     Max
-44.699 -10.591  -1.930   7.715 116.603

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        19.0691     2.9482   6.468 1.23e-10 ***
reconstr_PctHS18_24   4.5679     0.0836  54.640  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.39 on 2133 degrees of freedom
Multiple R-squared:  0.5833,    Adjusted R-squared:  0.5831
F-statistic:  2986 on 1 and 2133 DF,  p-value: < 2.2e-16

[[2]]

Call:
lm(formula = reconstr_TARGET_deathRate ~ ., data = training[,
    imp_feat[c(1:i, 7)], with = F])

Residuals:
    Min      1Q  Median      3Q     Max
-50.488  -8.974  -0.018   8.890 135.520

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               77.57561    3.99804   19.40   <2e-16 ***
reconstr_PctHS18_24        4.18410    0.07918   52.84   <2e-16 ***
reconstr_PctPrivateCoverage -0.70034    0.03521  -19.89   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.06 on 2132 degrees of freedom
Multiple R-squared:  0.6485,    Adjusted R-squared:  0.6482
F-statistic:  1967 on 2 and 2132 DF,  p-value: < 2.2e-16
```

```
[[3]]

Call:
lm(formula = reconstr_TARGET_deathRate ~ ., data = training[,
    imp_feat[c(1:i, 7)], with = F])

Residuals:
    Min      1Q  Median      3Q     Max
-50.696  -8.841  -0.262   8.553 134.412

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                76.04149    3.97067  19.151  < 2e-16 ***
reconstr_PctHS18_24         3.50227    0.13465  26.010  < 2e-16 ***
reconstr_PctPrivateCoverage -0.98342    0.05728 -17.168  < 2e-16 ***
reconstr_BirthRate          7.77854    1.24822   6.232 5.55e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.93 on 2131 degrees of freedom
Multiple R-squared:  0.6548,    Adjusted R-squared:  0.6543
F-statistic:  1347 on 3 and 2131 DF,  p-value: < 2.2e-16
```

```
[[4]]

Call:
lm(formula = reconstr_TARGET_deathRate ~ ., data = training[,
    imp_feat[c(1:i, 7)], with = F])

Residuals:
    Min      1Q  Median      3Q     Max
-41.035  -6.073  -0.048   5.977  65.053

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                19.15894    2.97792   6.434 1.53e-10 ***
reconstr_PctHS18_24         3.05223    0.09329  32.717  < 2e-16 ***
reconstr_PctPrivateCoverage -1.33146    0.04014 -33.173  < 2e-16 ***
reconstr_BirthRate          8.21947    0.86058   9.551  < 2e-16 ***
reconstr_incidenceRate      0.20620    0.00425  48.515  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.29 on 2130 degrees of freedom
Multiple R-squared:  0.836,     Adjusted R-squared:  0.8357
F-statistic:  2715 on 4 and 2130 DF,  p-value: < 2.2e-16
```

```
[[5]]

Call:
lm(formula = reconstr_TARGET_deathRate ~ ., data = training[,
    imp_feat[c(1:i, 7)], with = F])

Residuals:
    Min      1Q  Median      3Q     Max
-39.359  -5.903   1.143   5.904  59.926

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                13.504574   2.856157   4.728 2.41e-06 ***
reconstr_PctHS18_24         4.328504   0.122636  35.296  < 2e-16 ***
reconstr_PctPrivateCoverage -1.490489   0.039595 -37.644  < 2e-16 ***
reconstr_BirthRate          9.668256   0.823865  11.735  < 2e-16 ***
reconstr_incidenceRate      0.236002   0.004499  52.456  < 2e-16 ***
reconstr_PctHS25_Over       -1.444846   0.095870 -15.071  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.785 on 2129 degrees of freedom
Multiple R-squared:  0.8518,    Adjusted R-squared:  0.8515
F-statistic:  2448 on 5 and 2129 DF,  p-value: < 2.2e-16

[[6]]

Call:
lm(formula = reconstr_TARGET_deathRate ~ ., data = training[,
    imp_feat[c(1:i, 7)], with = F])

Residuals:
    Min      1Q  Median      3Q     Max
-47.393  -5.488   1.055   5.939  41.303

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                7.9450793  2.7771968   2.861  0.00427 **
reconstr_PctHS18_24        4.4514762  0.1182453  37.646  < 2e-16 ***
reconstr_PctPrivateCoverage -1.4379412  0.0382649 -37.579  < 2e-16 ***
reconstr_BirthRate         8.1029444  0.8006582  10.120  < 2e-16 ***
reconstr_incidenceRate     0.2239555  0.0044188  50.682  < 2e-16 ***
reconstr_PctHS25_Over      -1.1408412  0.0949566 -12.014  < 2e-16 ***
reconstr_avgAnnCount       0.0025284  0.0001905  13.274  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.406 on 2128 degrees of freedom
Multiple R-squared:  0.8632,    Adjusted R-squared:  0.8628
F-statistic:  2237 on 6 and 2128 DF,  p-value: < 2.2e-16
```

After looking at all 6 models, we see that the 6th model gives us the highest adjusted R^2 and all the variables are significant. Next, we will check the VIF to verify there is no multi-collinearity in the model.

FINAL MODEL:

```
> summary(model1)

Call:
lm(formula = reconstr_TARGET_deathRate ~ reconstr_PctHS18_24 +
    reconstr_PctPrivateCoverage + reconstr_BirthRate + reconstr_incidenceRate +
    reconstr_PctHS25_Over + reconstr_avgAnnCount, data = training)

Residuals:
    Min      1Q  Median      3Q     Max
-47.393  -5.488   1.055   5.939  41.303

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                7.9450793  2.7771968   2.861  0.00427 **
reconstr_PctHS18_24        4.4514762  0.1182453  37.646  < 2e-16 ***
reconstr_PctPrivateCoverage -1.4379412  0.0382649 -37.579  < 2e-16 ***
reconstr_BirthRate         8.1029444  0.8006582  10.120  < 2e-16 ***
reconstr_incidenceRate     0.2239555  0.0044188  50.682  < 2e-16 ***
reconstr_PctHS25_Over      -1.1408412  0.0949566 -12.014  < 2e-16 ***
reconstr_avgAnnCount       0.0025284  0.0001905  13.274  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.406 on 2128 degrees of freedom
Multiple R-squared:  0.8632,    Adjusted R-squared:  0.8628
F-statistic:  2237 on 6 and 2128 DF,  p-value: < 2.2e-16
```

```
> vif(model1)
      reconstr_PctHS18_24 reconstr_PctPrivateCoverage          reconstr_BirthRate
                 4.077921                   3.219738                    4.070149
   reconstr_incidenceRate       reconstr_PctHS25_Over        reconstr_avgAnnCount
                 1.409457                   4.554474                    1.310346
```

After checking the VIF, we can see that all the VIF's are less than 5, which suggests that there is no multicollinearity present in the model. The adjusted R^2 is 86.28%, which is fairly high when compared to 69.09% which we were getting from the StepAIC method before performing feature engineering and using the GLRM model. This is our final model, and we will next check whether the model is violating any assumption of multiple linear regression.

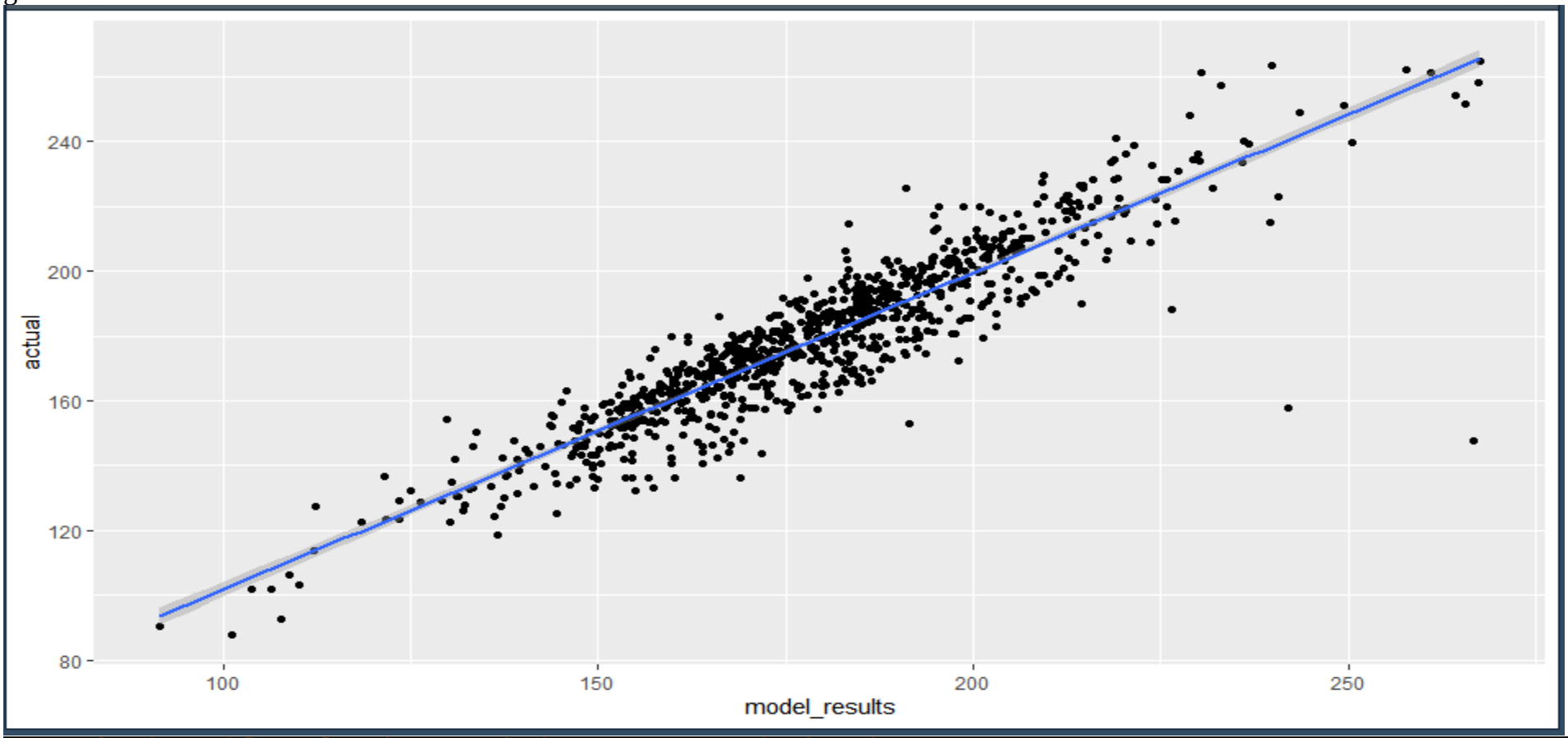
After conducting the summary and model analysis for the final model, we found out some of the findings which are as follows:

- The p-values for all the variables are significant at the 95% significance level.
- The VIF values for all variables in this model are less than the selected threshold, 5. This means we can use these variables in the same model since it avoids the problem of multi-collinearity.
- The adjusted R-squared value was 0.8628, which is higher than any of the previous model.

Therefore, our final model comprised of reconstr_PctHS18_24, reconstr_PctPrivateCoverage, reconstr_BirthRate, reconstr_incidenceRate, reconstr_PctHS25_Over, reconstr_avgannCount.

MODEL PREDICTION:

After a model has been processed by using the training set, you test the model by making predictions against the test set. Because the data in the training set already contains known values for the attribute that you want to predict, it is easy to determine whether the model's guesses are correct.

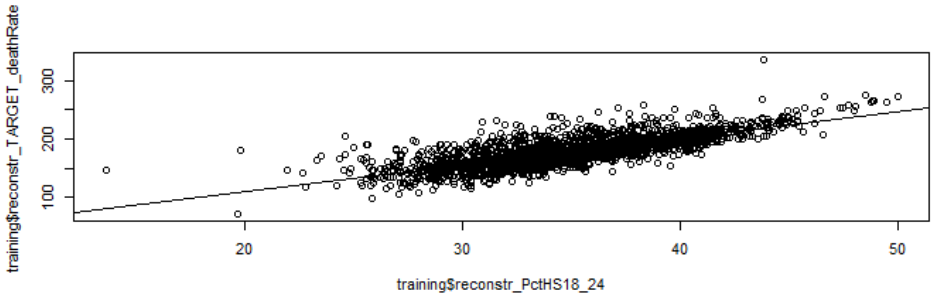


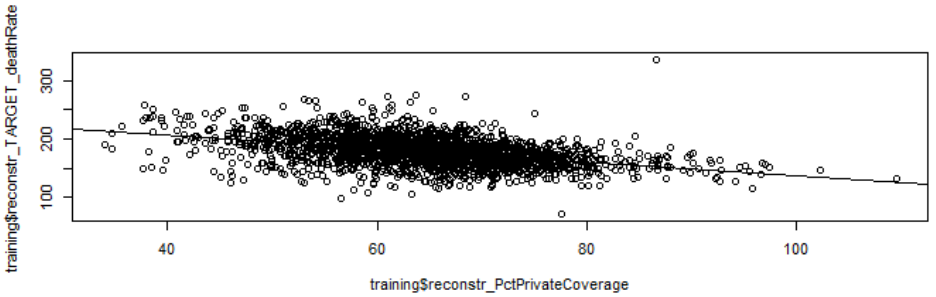

```
> RMSE(pred = model_results,obs = testing$reconstr_TARGET_deathRate)
[1] 10.61335
>
```

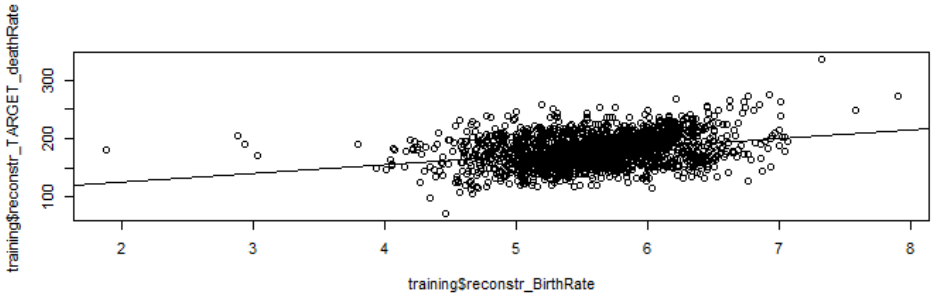
The test set was scored on the model and the result imply satisfactory results when comparing the predicted values with the expected result having a RMSE of 10.61335. The residual standard error of the training model was also 9.406, which shows the prediction is very accurate.

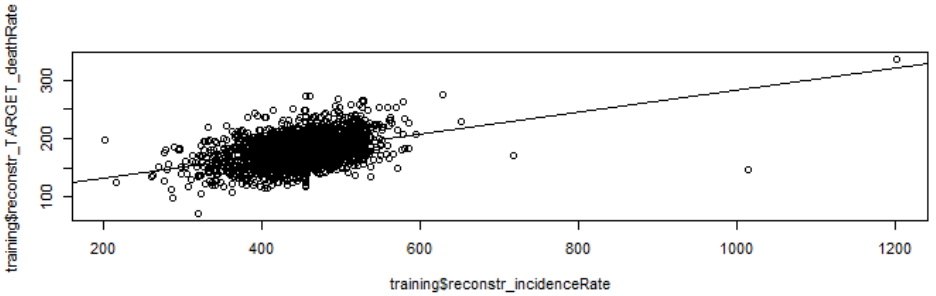
CHECKING THE ASSUMPTION OF MULTIPLE LINEAR REGRESSION

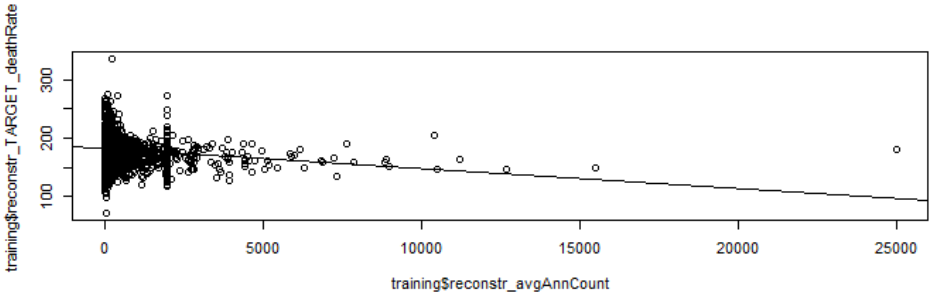
1. **Linearity**: From the scatter plot of selling price vs predictor variables we see that there is a linear relationship. Hence, the linearity assumption is not violated.







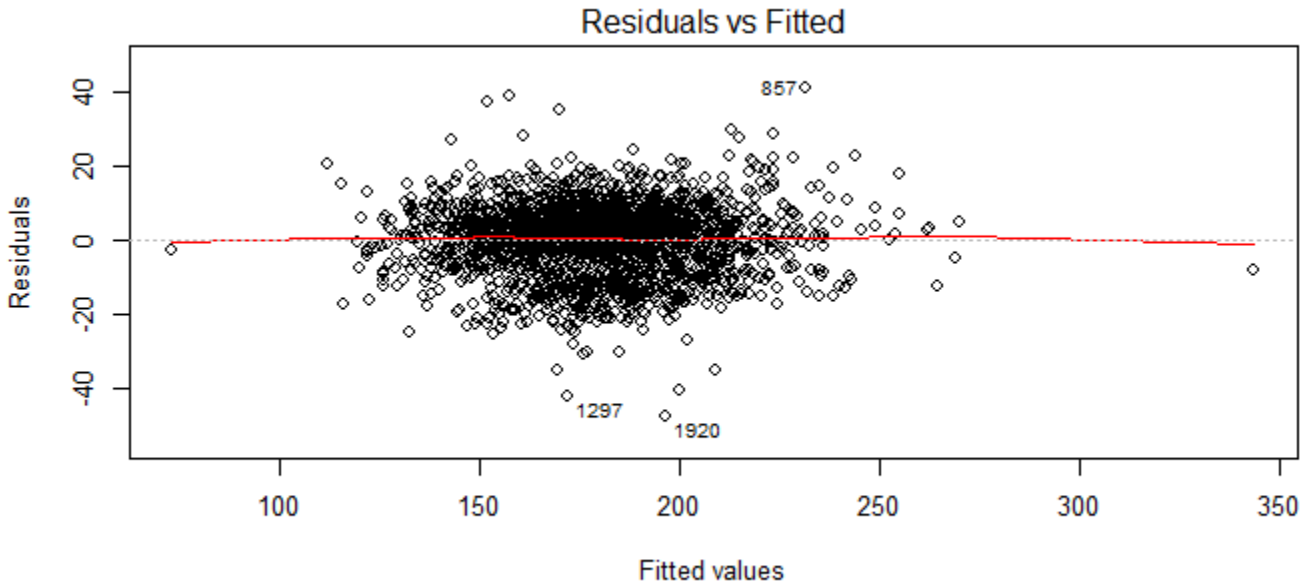




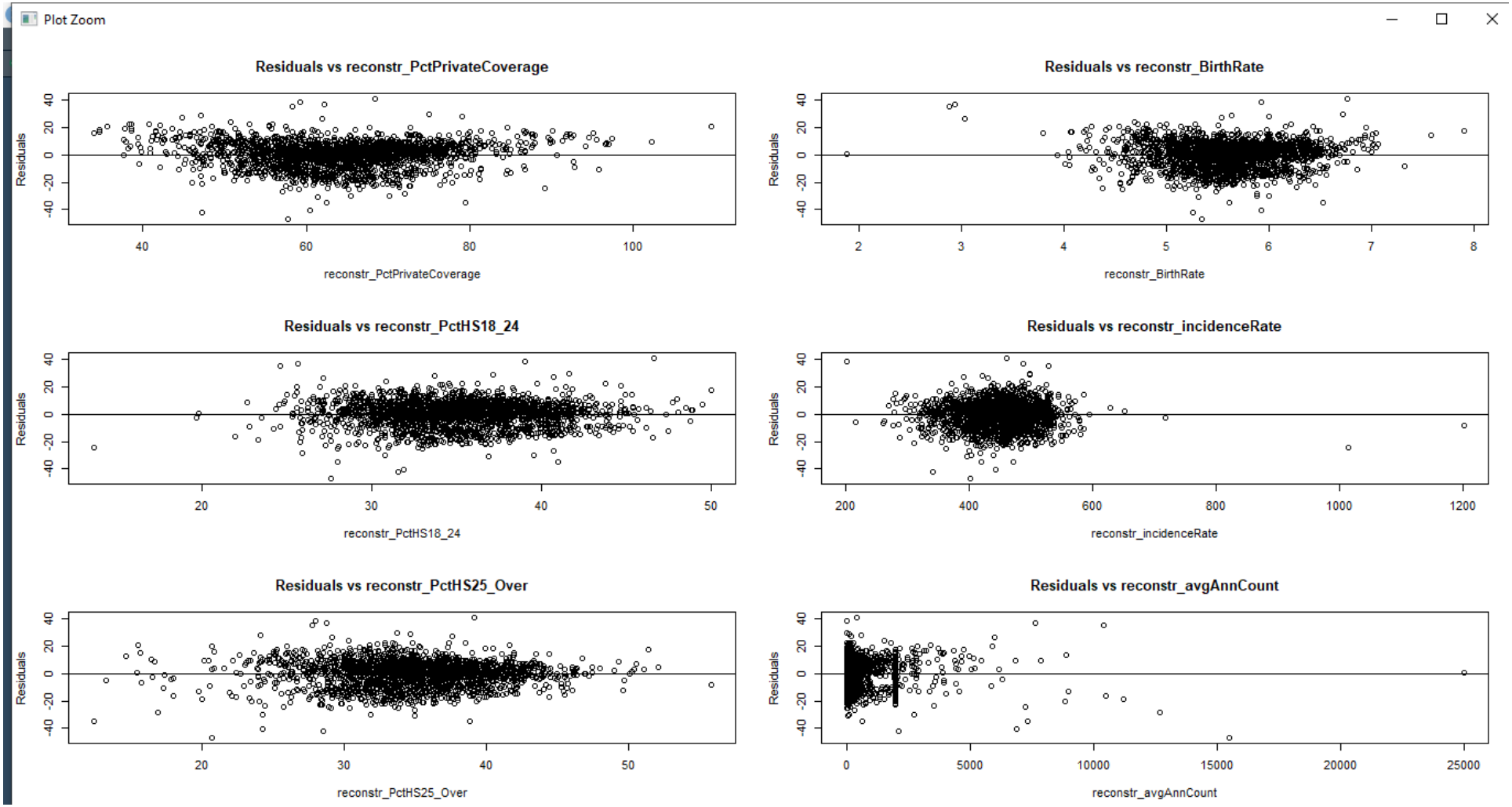
2. **The model errors are independent**: From the DW Test, we get the value of 1.8381, which is close to 2. The closer the value to 2, it means no autocorrelation. Since the p-value is 0.2549, which is more than 0.05, it is safe to say that there is no autocorrelation in the model. This assumption is not violated.

```
        Durbin-Watson test

data:  model1
DW = 1.8381, p-value = 0.2549
alternative hypothesis: true autocorrelation is greater than 0
```

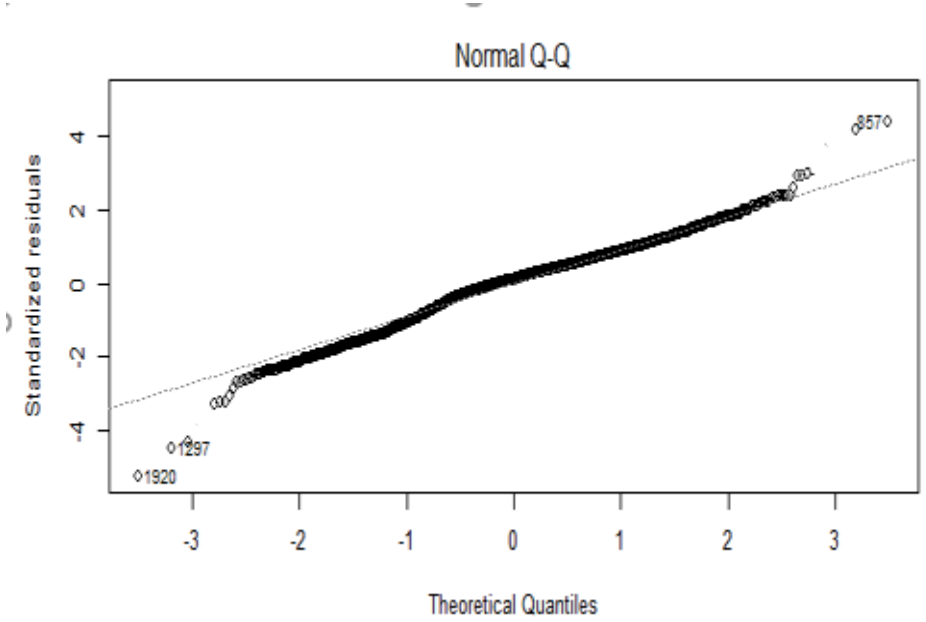
3. **Homoscedasticity/Errors have a constant variance**: From the residual vs. fitted plot, we can see that the errors are distributed randomly and there is no apparent pattern. This assumption is not violated.

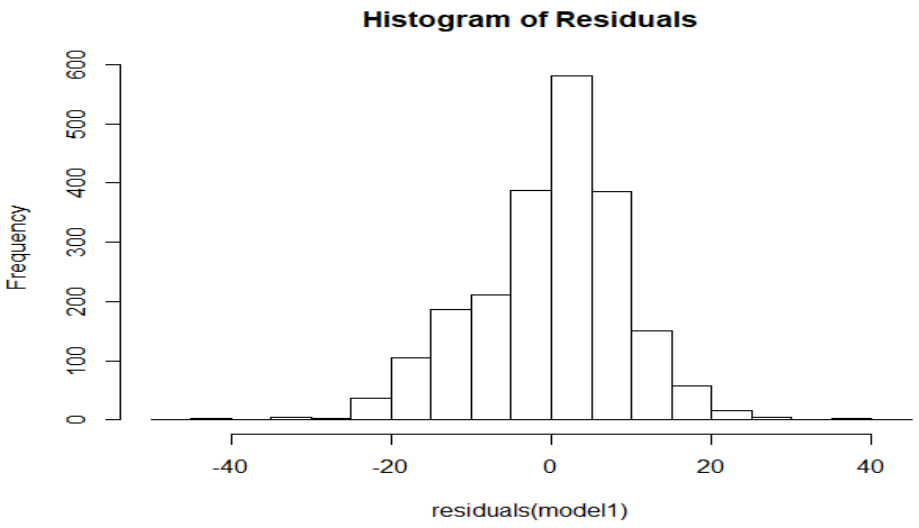


4. Independent variables: need to be independent variables: The residual vs. independent variables plot shows no pattern and it is random and constant, the VIF was also less than 5. This indicates that the assumption has not been violated.



5. The errors are normally distributed: The normal Q-Q plot shows a normal distribution of that and to validate this assumption we also looked at the histogram of the residuals which shows the data are normally distributed. Hence,this assumption is not violated.





Therefore, none of the assumptions have been violated by this model and it is meeting all the five assumption of the Multiple Linear Regression. This is the final, validated model!

Ridge Regression

```
#Ridge regression

ridge_mod = glmnet(xtrainmat,
                   ytrainmat,
                   alpha = 0) # Fit lasso model on training data

plot(ridge_mod)

ridge=cv.glmnet(xtrainmat,ytrainmat,alpha=0)
par(mfrow=c(1,1))
plot(ridge)

bestlam = ridge$lambda.min # Select lamda that minimizes training MSE
ridge_pred = predict(ridge_mod, s = bestlam, newx = xtestmat) # Use best lambda to predict test data
mean((ridge_pred - y_test)^2) # Calculate test MSE
RMSE(pred = ridge_pred,obs = y_test)
```

In ridge regression, we add a penalty by way of a tuning parameter called lambda which is chosen using cross validation. The idea is to make the fit small by making the residual sum or squares small plus adding a shrinkage penalty. The shrinkage penalty is lambda times the sum of squares of the coefficients so coefficients that get too large are penalized. As lambda gets larger, the bias is unchanged, but the variance drops. The drawback of ridge is that it does not select variables. It includes all of the variables in the final model

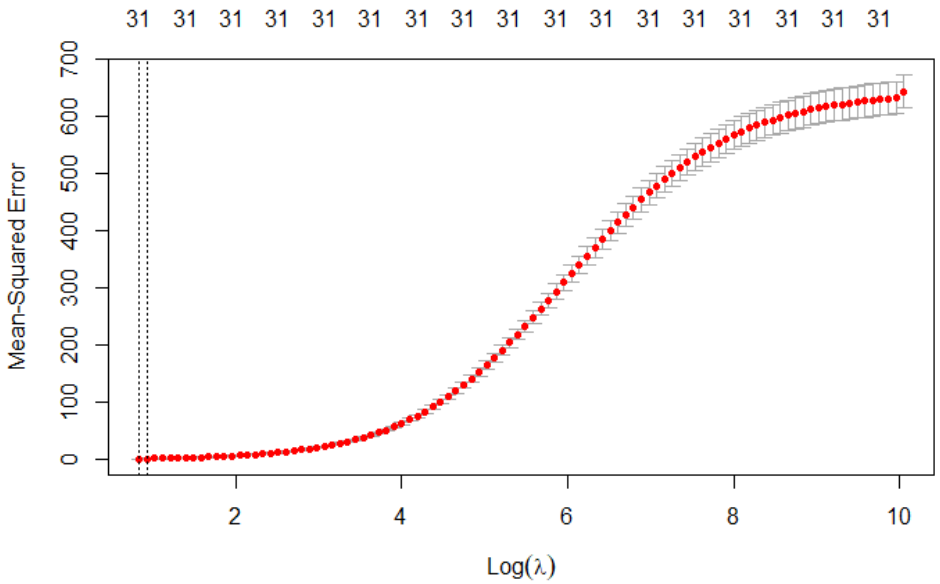
LASSO REGRESSION

```
#lasso regression
install.packages('glmnet')
library(glmnet)
xtrainmat <- as.matrix(x_train)
ytrainmat=as.matrix(y_train)
xtestmat=as.matrix(x_test)
lasso_mod = glmnet(xtrainmat,
                   ytrainmat,
                   alpha = 1) # Fit lasso model on training data

plot(lasso_mod)
lasso=cv.glmnet(xtrainmat,ytrainmat,alpha=1)
par(mfrow=c(1,1))
plot(lasso)

bestlam = lasso$lambda.min # Select lamda that minimizes training MSE
lasso_pred = predict(lasso_mod, s = bestlam, newx = xtestmat) # Use best lambda to predict test data
mean((lasso_pred - y_test)^2) # Calculate test MSE
RMSE(pred = lasso_pred,obs = y_test)


out = glmnet(xtrainmat, ytrainmat, alpha = 1) # Fit lasso model on full dataset
lasso_coef = predict(out, type = "coefficients", s = bestlam)[1:20,] # Display coefficients using lambda chosen
lasso_coef
```

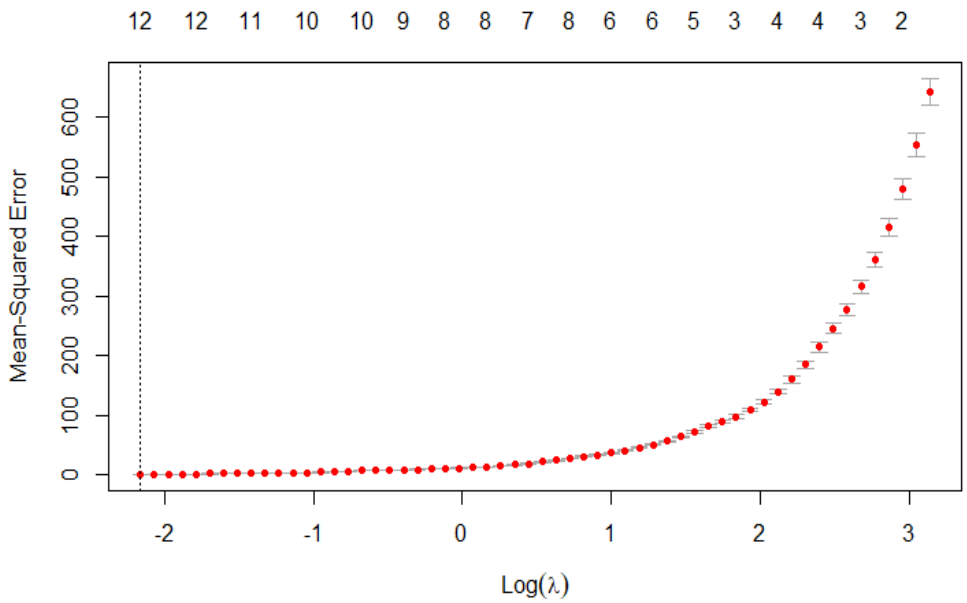
In lasso, the penalty is the sum of the absolute values of the coefficients. Lasso shrinks the coefficient estimates towards zero and it has the effect of setting variables exactly equal to zero when lambda is large enough while ridge does not. Hence, much like the best subset selection method, lasso performs variable selection. The tuning parameter lambda is chosen by cross validation. When lambda is small, the result is essentially the least squares estimates. As lambda increases, shrinkage occurs so that variables that are at zero can be thrown away. So, a major advantage of lasso is that it is a combination of both shrinkage and selection of variables. In cases with very large number of features, lasso allow us to efficiently find the sparse model that involve a small subset of the features.

```
> out = glmnet(xtrainmat, ytrainmat, alpha = 1) # Fit lasso model on full dataset
> lasso_coef = predict(out, type = "coefficients", s = bestlam)[1:20,] # Display coefficients using lambda chosen by CV
> lasso_coef
                  (Intercept)          reconstr_avgAnnCount   reconstr_avgDeathsPerYear       reconstr_incidenceRate
                 0.3424425166                  0.0002148411                  0.0000000000                  0.0000000000
          reconstr_medIncome             reconstr_popEst2015     reconstr_povertyPercent          reconstr_studyPerCap
                 0.0000000000                  0.0000000000                  0.0000000000                  0.0031400908
         reconstr_MedianAge       reconstr_MedianAgeMale     reconstr_MedianAgeFemale   reconstr_AvgHouseholdSize
                -0.0666014922                  0.0000000000                  1.7366102410                  0.0000000000
    reconstr_PercentMarried          reconstr_PctNoHS18_24          reconstr_PctHS18_24     reconstr_PctSomeCol18_24
                 0.0000000000                  2.6532450077                  0.0000000000                 -0.1183875607
   reconstr_PctBachDeg18_24          reconstr_PctHS25_Over reconstr_PctBachDeg25_Over reconstr_PctEmployed16_Over
                 0.0000000000                  0.4112282297                  0.0000000000                  0.0000000000
>
```

RIDGE REGRESSION PLOT



LASSO REGRESSION PLOT




```
> mean((ridge_pred - y_test)^2) # Calculate test MSE
[1] 3.550543
> RMSE(pred = ridge_pred,obs = y_test)
[1] 1.884288
>
```

```
> mean((lasso_pred - y_test)^2) # Calculate test MSE
[1] 1.166031
> RMSE(pred = lasso_pred,obs = y_test)
[1] 1.079829
>
```

```
> bestlamridge
[1] 2.312919
> bestlamlasso
[1] 0.1151145
>
```

By comparing the Ridge and Lasso regression plots, we can see that at smaller values of lambda for Lasso regression, the mean squared error is lower when compare to ridge regression. The best lambda value was found to be 2.31 for ridge regression whereas it was 0.11 for Lasso regression. The RMSE is also lower for Lasso regression, 1.0798 when compared to 1.8843 of Ridge regression.

RANDOM FOREST

Random Forest algorithm is one of the most commonly used and the most powerful machine learning techniques. It is a special type of bagging applied to decision trees.

```
#random forest
install.packages('randomForest')
library(randomForest)
require(caTools)
install.packages('quantmod')
library(quantmod)


rf <- randomForest(
  y_train ~ .,
  data=x_train
)

pred_rf = predict(rf, newdata=x_test)
pred_rf
RMSE(pred = pred_rf,obs = y_test)
postResample(pred_rf, y_test)
compare_rf <- cbind (actual=y_test, pred_rf)# combine
compare_rf
#calculate accuracy
mean (apply(compare_rf, 1, min)/apply(compare_rf, 1, max))
```

```
> RMSE(pred = pred_rf,obs = y_test)
[1] 5.675191
```

Using random forest regression, we have computed the prediction error, which is measured by RMSE, which corresponds to the average difference between the observed known values of the outcome and the predicted value by the model. The lower the RMSE, the better the model. And for our data set, the RMSE value is showing 5.675191 which indicates the absolute fit of the model to the data

Cross Validation – Leave one out Cross validation

```
#cross validation --- LOOCV
# Define training control
train.control <- trainControl(method = "LOOCV")
# Train the model
model <- train(reconstr_TARGET_deathRate ~., data = canc2, method = "lm",
               trControl = train.control)
# Summarize the results
print(model)
```

```
> # Summarize the results
> print(model)
Linear Regression

3047 samples
  31 predictors

No pre-processing
Resampling: Leave-One-Out Cross-Validation
Summary of sample sizes: 3046, 3046, 3046, 3046, 3046, 3046, ...
Resampling results:

  RMSE      Rsquared   MAE
  12.23694  0.8208429  10.72714


Tuning parameter 'intercept' was held constant at a value of TRUE
```

The advantage of the LOOCV method is that we make use all data points reducing potential bias.However, the process is repeated as many times as there are data points, resulting to a higher execution time when n is extremely large.Additionally, we test the model performance against one data point at each iteration.

Using the LOOCV method, we train the model on the data and the output shows us that the RMSE is 12.236 and the R2 value is 82.08%, which when we compare with our final model R2 of 86.28%, we can say that the model is not overfitting and the bias vs. variance tradeoff is met.