County Based Opportunites for Children

BQOM 2578 | Data Mining | Final Project

Group 8: Anthony Pulleo, Hannah Shernisky, Theresa Wohlever

Thursday, December 4, 2025

Table of contents

Executive Summary

Group 8: Anthony Pulleo, Hannah Shernisky, Theresa Wohlever

Data Preparation

The Child Opportunity Index (COI) measures and maps the quality of resources and conditions like these that matter for children's healthy development in the neighborhoods where they live.

Importing Data, Cleaning, & Wrangling

The data from diversitydatakids.org contains a series of indices, and does not provide the raw data that was used in the index calculation. The indices are normalized across different areas, like education or housing. We will pull from other datasets to see if factors calculated / assessed by those datasets influence the COI.

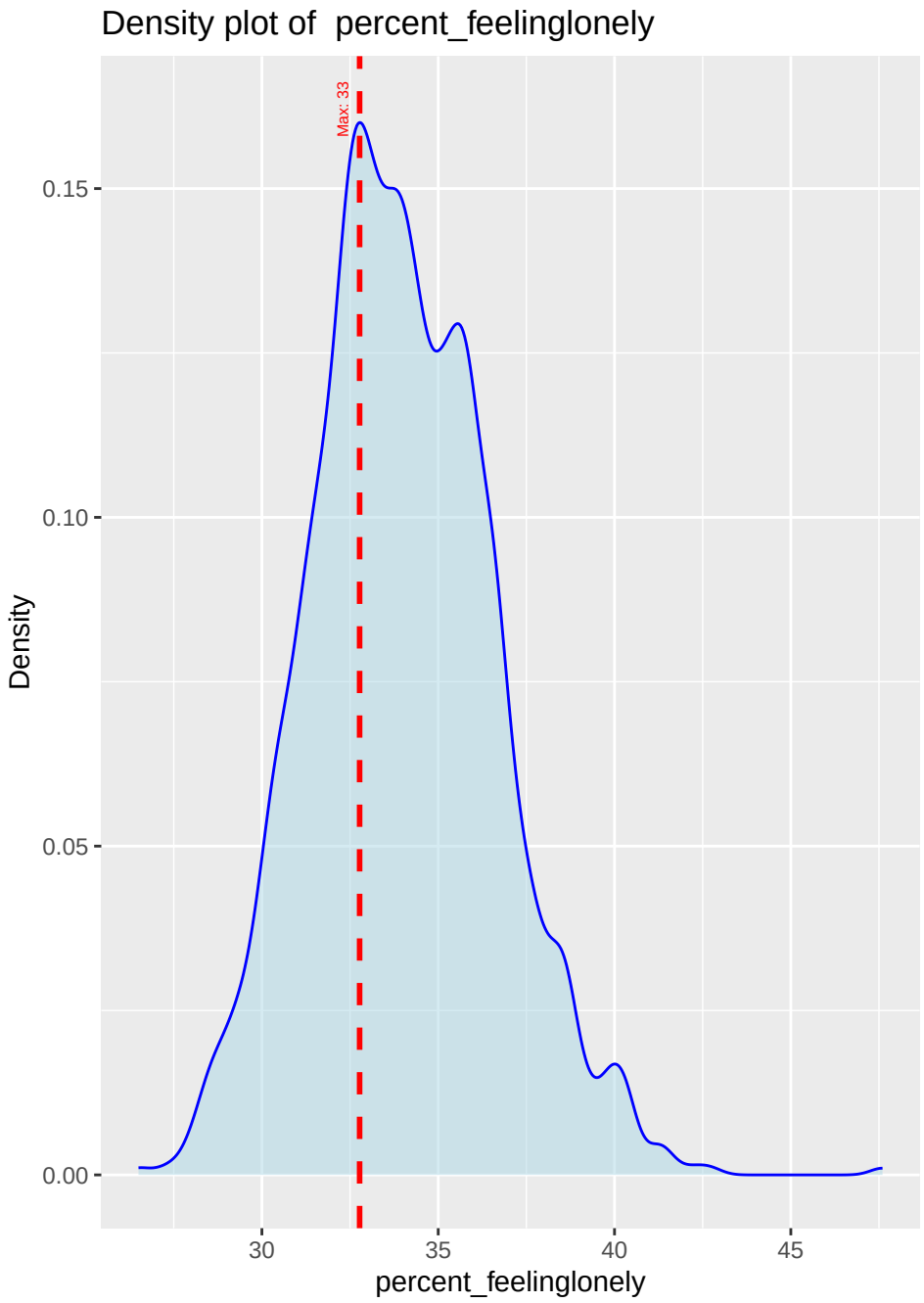
These datasets include the Urban Influence Codes from USDA, Census Data as part of the American Community Survey, and the 2025 County Health Rankings Data. Select variables will be appended via a left_join by County FIPS codes.

Data Exploration

Identify dimensions of interest

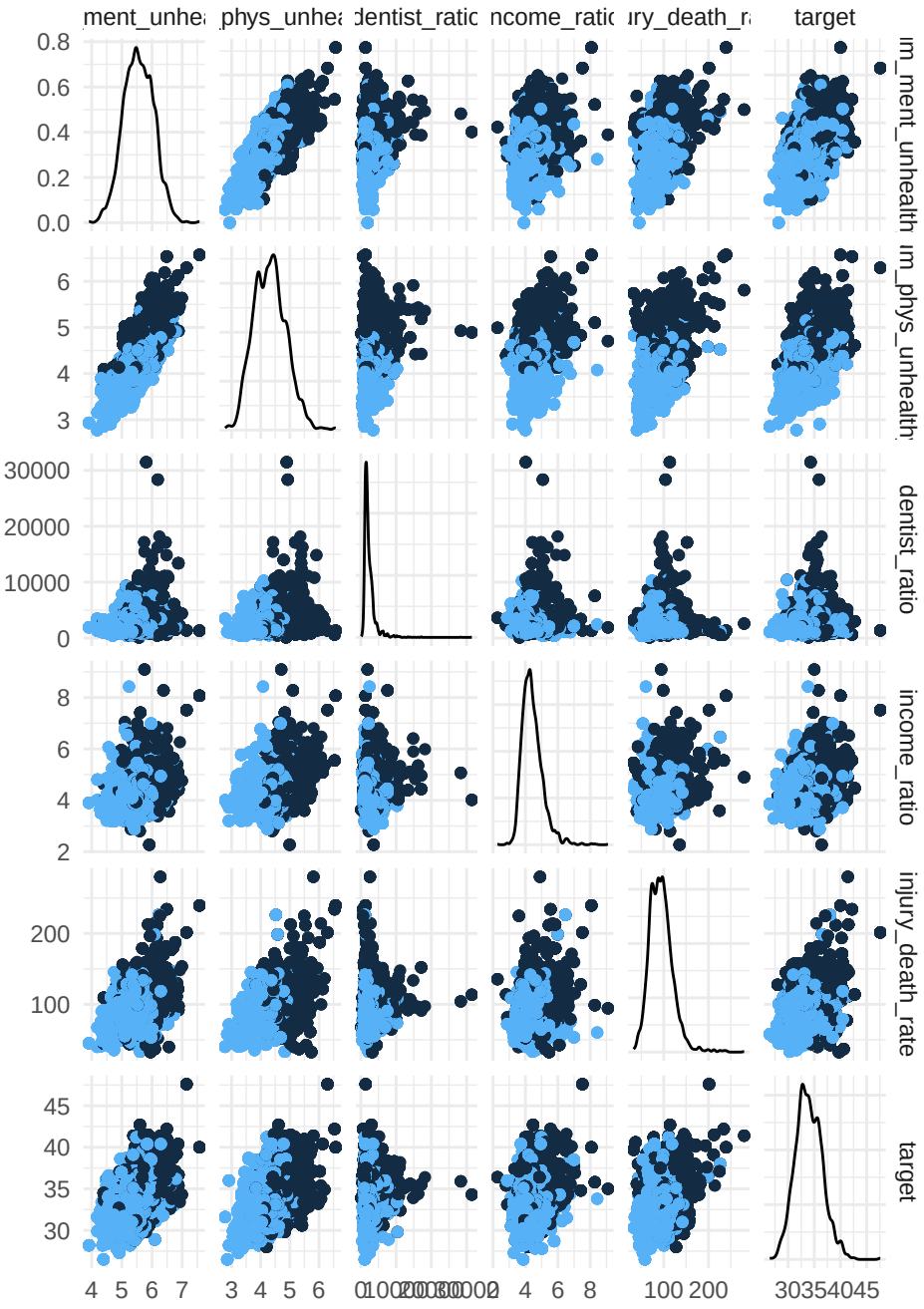
```
df <- df_alljoined


#
# COI
#
# Histogram of target values
coi_density <- density(df$r_COI_nat)

# Convert the density estimate to a function
coi_dens_func <- approxfun(coi_density$x, coi_density$y)

# Use optimize() to find the maximum in a specified interval (choose based on your data)
coi_result <- optimize(coi_dens_func, interval = c(min(df$r_COI_nat), max(df$r_COI_nat)), maximum = TRUE)
coi_local_max_x <- coi_result$maximum     # The x value where local max occurs
```

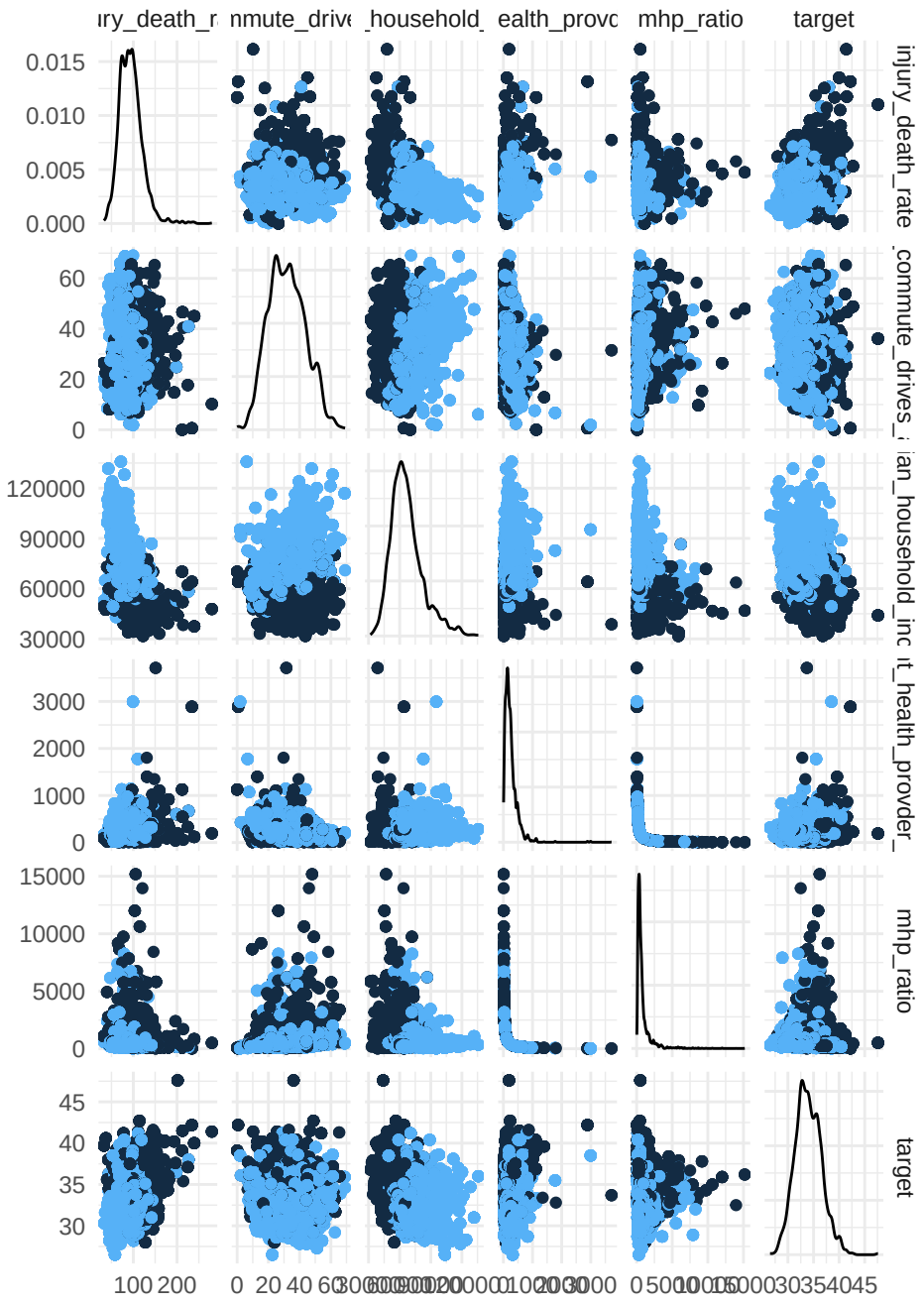
```r
coi_local_max_y <- coi_result$objective   # The max density value

## Make COI binary
coi_bin_cutoff <- coi_local_max_x
df$coi_bin <- ifelse(df$r_COI_nat < coi_bin_cutoff, 0, 1)




###
### TARGET
###

# df$target <- df$r_COI_nat
df$target <- df[[target_name]]
df[[target_name]] <- NULL
colsOfInterest <- colsOfInterest[!(colsOfInterest %in% target_name)]
colsOfInterest <- c(colsOfInterest, "target")

##
## Visualize target values
##

# Histogram of target values
target_density <- density(df$target)

# Convert the density estimate to a function
dens_func <- approxfun(target_density$x, target_density$y)

# Use optimize() to find the maximum in a specified interval (choose based on your data)
result <- optimize(dens_func, interval = c(min(df$target), max(df$target)), maximum = TRUE)
local_max_x <- result$maximum     # The x value where local max occurs
local_max_y <- result$objective   # The max density value

#  Create density plot with ggplot2 and add vertical line at max
df_density <- data.frame(x = df$target)
ggplot(df_density, aes(x = df$target)) +
  geom_density(fill = "lightblue", color = "blue", alpha = 0.5) +
  geom_vline(xintercept = local_max_x , color = "red", linetype = "dashed", size = 1) +
  annotate("text", x = local_max_x, y = local_max_y + 0.002,
    label = sprintf("Max: %.0f", local_max_x), color = "red", angle = 90, vjust = -1, size = 2) +
  labs(title = paste("Density plot of ", target_name), x = target_name, y = "Density")
```

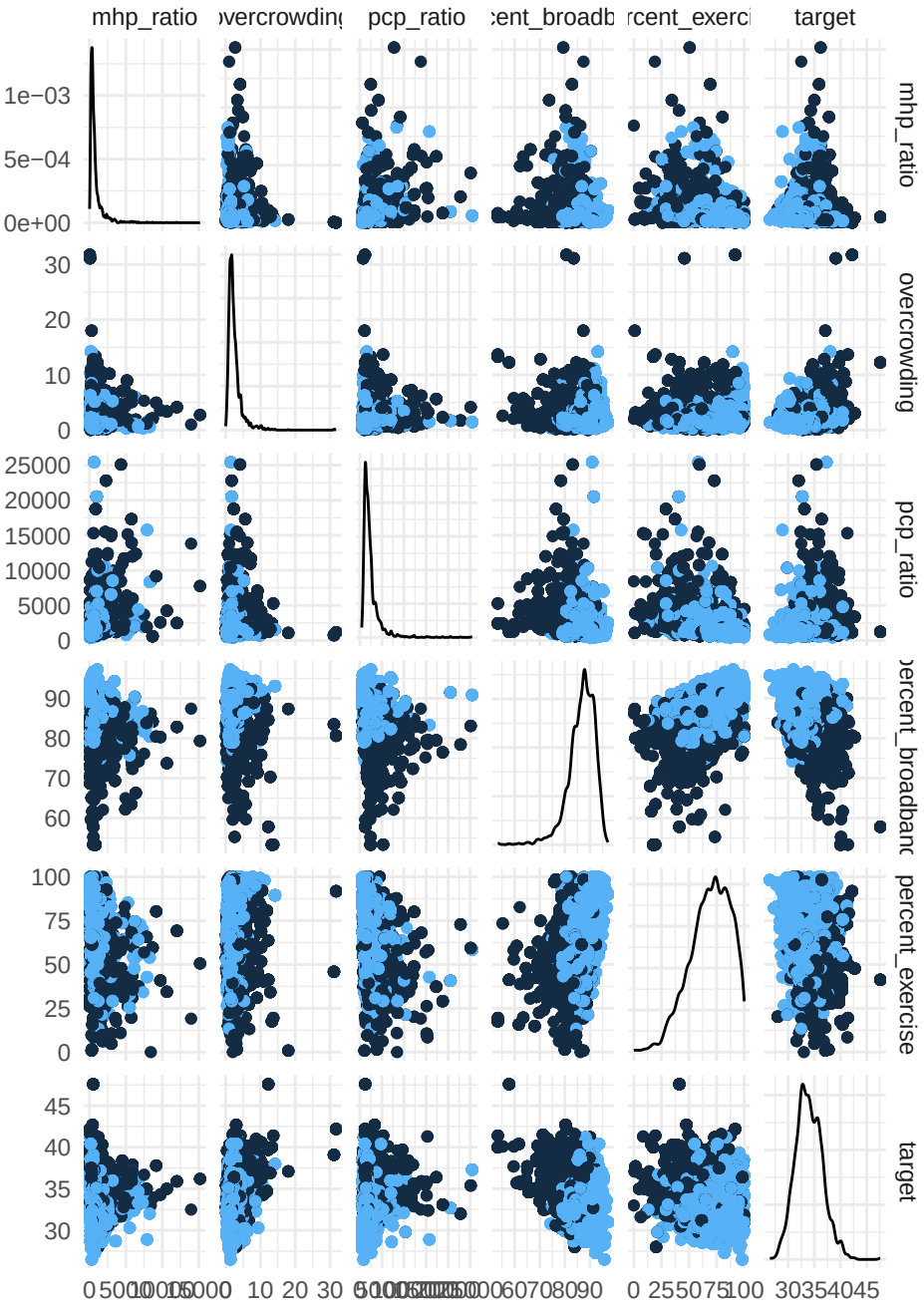


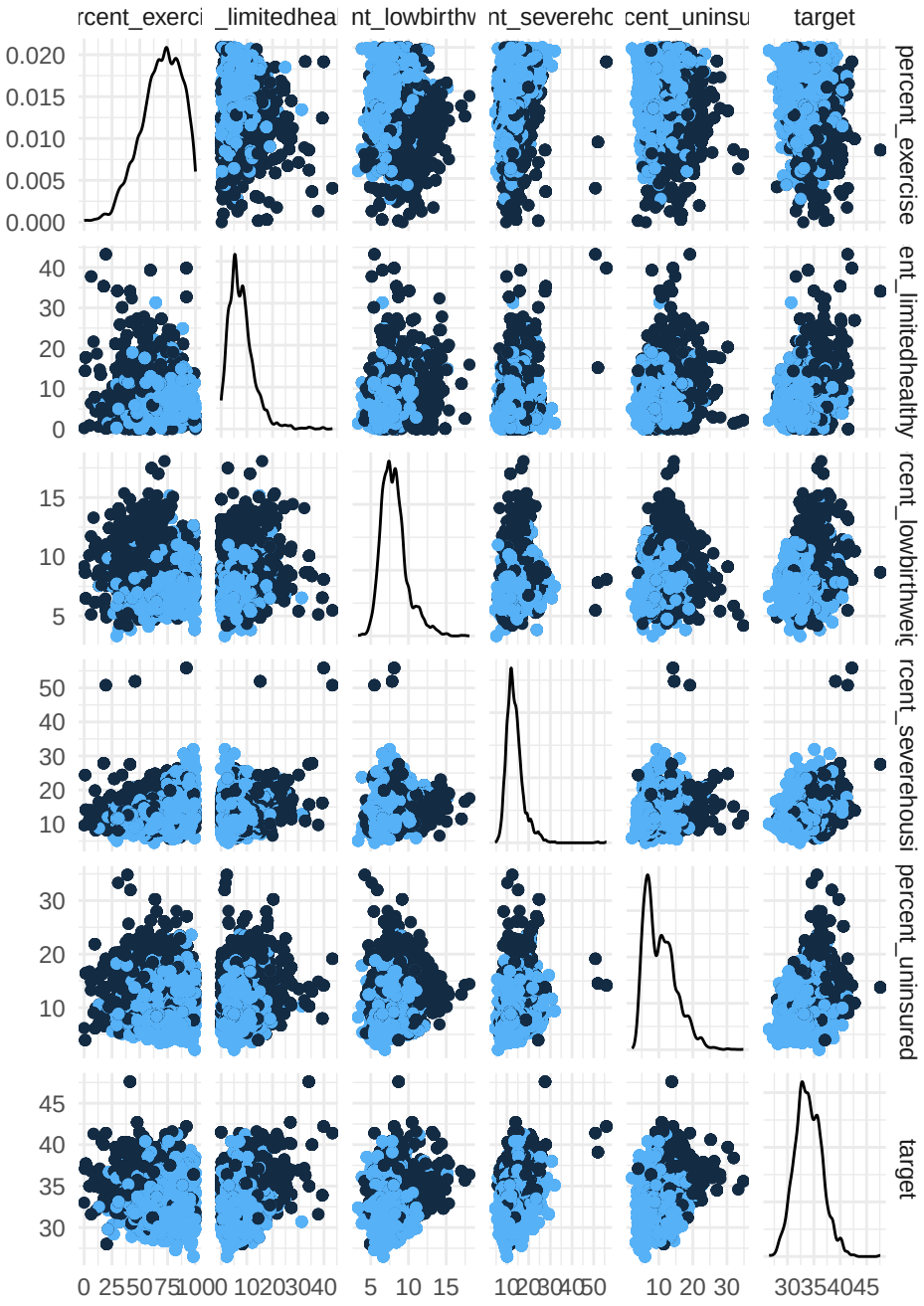

```
## Make TARGET binary
target_bin_cutoff <- local_max_x
df$target_bin <- ifelse(df$target < target_bin_cutoff, 0, 1)
```

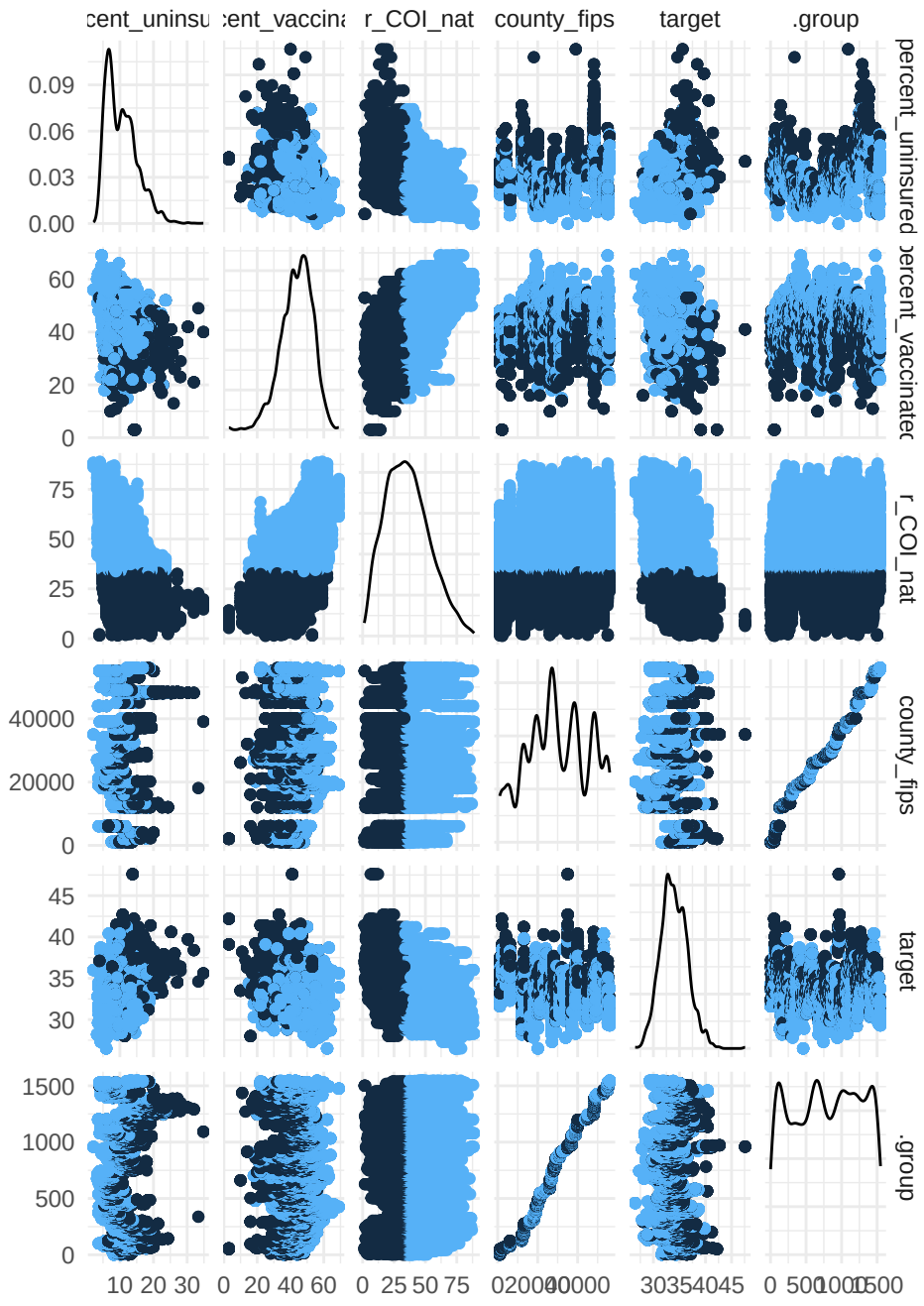
```r
## Reasonably sized scatter plots
bySeq <- 4
modSeq <- length(colsOfInterest) %% bySeq
cols2scatter <- colsOfInterest[!(colsOfInterest %in% "target")]
scatterSeq <- seq.int(1, length(colsOfInterest), bySeq)

for(c in scatterSeq)
{
  if(c == scatterSeq[length(scatterSeq)]){bySeq <- bySeq + modSeq}
  cols2Plot <- unique(c(colsOfInterest[c:(c+bySeq)], "target"))
  cols2Plot <- cols2Plot[!is.na(cols2Plot)]
  if(2 > length(cols2Plot)){next}

  scatter_plot_matrix <- ggpairs(
    df[ , cols2Plot],
    aes(color = df$coi_bin),
    upper = list(continuous = "points"),
    lower = list(continuous = "points"),
    diag  = list(continuous = "densityDiag")
  ) +
    theme_minimal()

  print(scatter_plot_matrix)
}
```


ment_unhea
phys_unhea
dentist_ratio
ncome_ratio
ury_death_r
target
m_ment_unhealth
m_phys_unhealth
dentist_ratio
income_ratio
injury_death_rate
target


ıry_death_r
nmute_drive
household
ealth_provc
mhp_ratio
target
injury_death_rate
commute_drives_
ian_household_inc
lt_health_provder_
mhp_ratio
target


mhp_ratio
overcrowding
pcp_ratio
cent_broadb
rcent_exerci
target
1e−03
5e−04
0e+00
30
20
10
0
25000
20000
15000
10000
5000
0
90
80
70
60
100
75
50
25
0
45
40
35
30
0 5000 10000 15000
0 10 20 30
0 25 50 75 100
30 35 40 45


rcent_exerci
_limitedhea
nt_lowbirthv
nt_severehc
cent_uninsu
target
percent_exercise
ent_limitedhealthy
rcent_lowbirthweig
rcent_severehousi
percent_uninsured
target



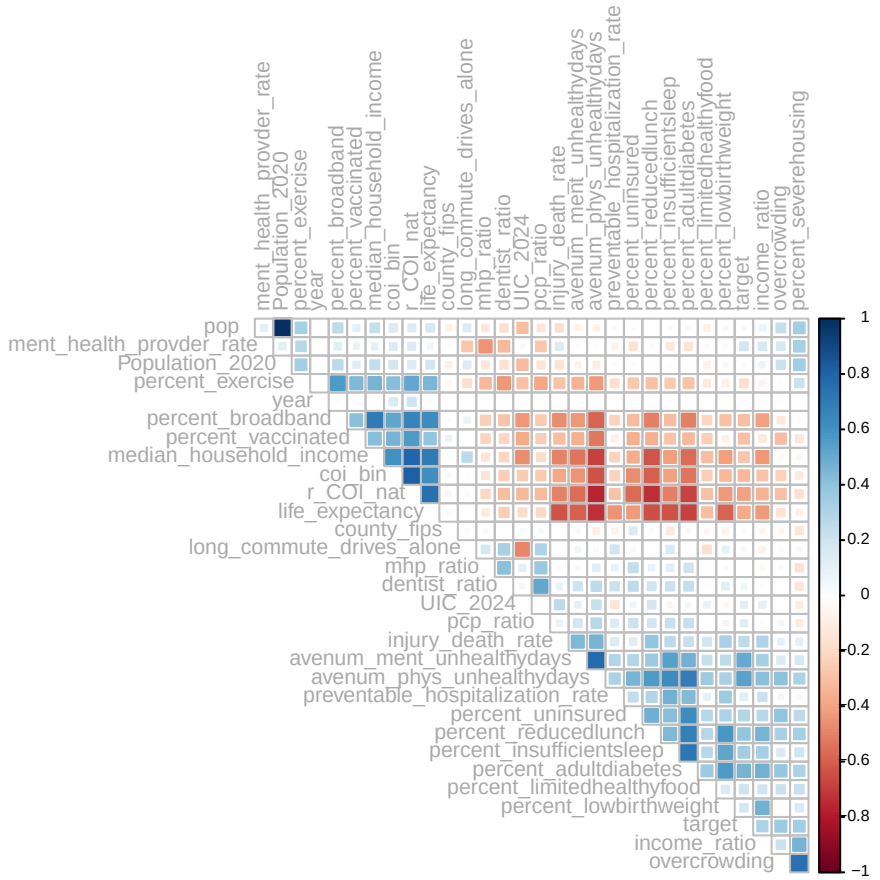

```
df_wTargetBin <- df
df$target_bin <- NULL
```

```
## Prep for correlation
df_cor <- df
```

```r
cor_mat <- cor(df %>% select(where(is.numeric)))
cor_threshold <- 0
cor_threshold_count <- 2

cols_above_threshold <- which( colSums(abs(cor_mat) > cor_threshold, na.rm = TRUE) >= cor_threshold_count)

df <- subset(df, select = colnames(cor_mat)[cols_above_threshold] )
cor_mat <- cor(df %>% select(where(is.numeric)))

cat(paste(colnames(cor_mat)[cols_above_threshold], collapse = "\n"))
```

```
county_fips
year
pop
r_COI_nat
UIC_2024
Population_2020
median_household_income
avenum_phys_unhealthydays
percent_lowbirthweight
avenum_ment_unhealthydays
ment_health_provder_rate
percent_vaccinated
percent_exercise
preventable_hospitalization_rate
pcp_ratio
mhp_ratio
dentist_ratio
percent_uninsured
percent_severehousing
percent_broadband
overcrowding
long_commute_drives_alone
income_ratio
injury_death_rate
life_expectancy
percent_adultdiabetes
percent_limitedhealthyfood
percent_insufficientsleep
percent_reducedlunch
coi_bin
target
```

```r
cor_mat_plot <- round(cor_mat, 2)
cor_mat_plot[is.na(cor_mat_plot)] <- 0 # Replace all NA values with zero
corrplot(cor_mat_plot,
  method="square",
  type="upper",
  order="AOE",
  tl.col="darkgrey",
  tl.cex = 0.7,      # label size
  cl.align.text = "r",
  cl.cex = 0.5,   # legend text
  diag=FALSE,
  number.cex=0.6)
```

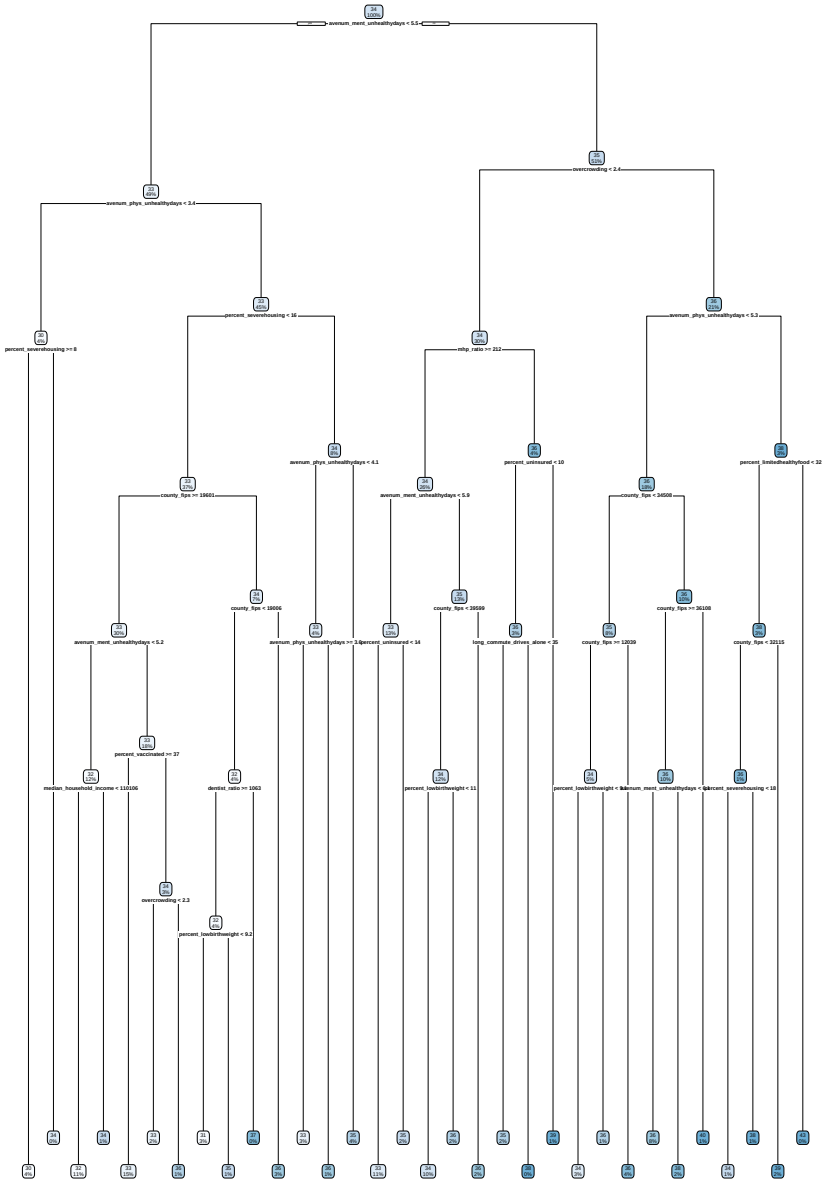


Modeling

Split data into training and testing

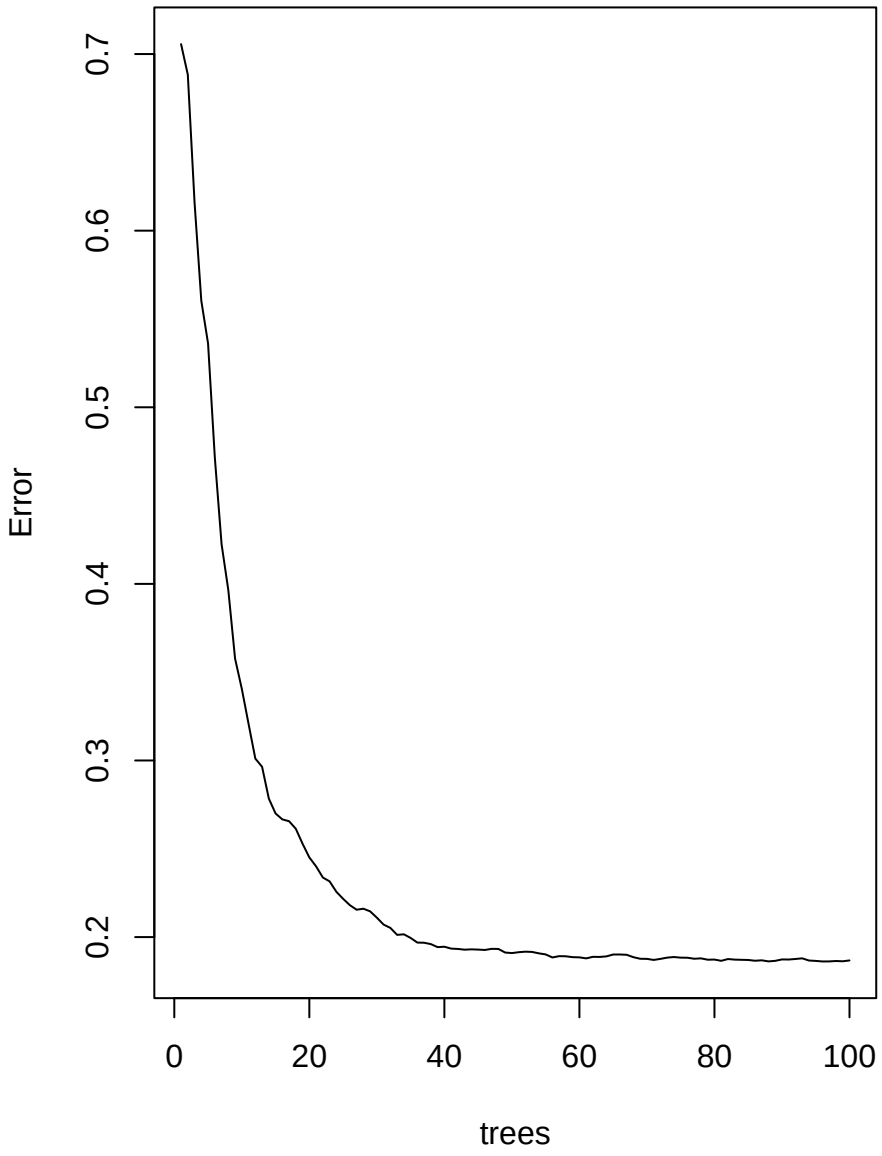
We will deviate slightly from the dataset we will use in our final project for the purpose of this homework. When doing the ANN, we identified a few variables that made the ANN return NaNs, and therefore inhibited our ability to successfully run these models. We will create df in the code below with variables that allow the ANN to work.

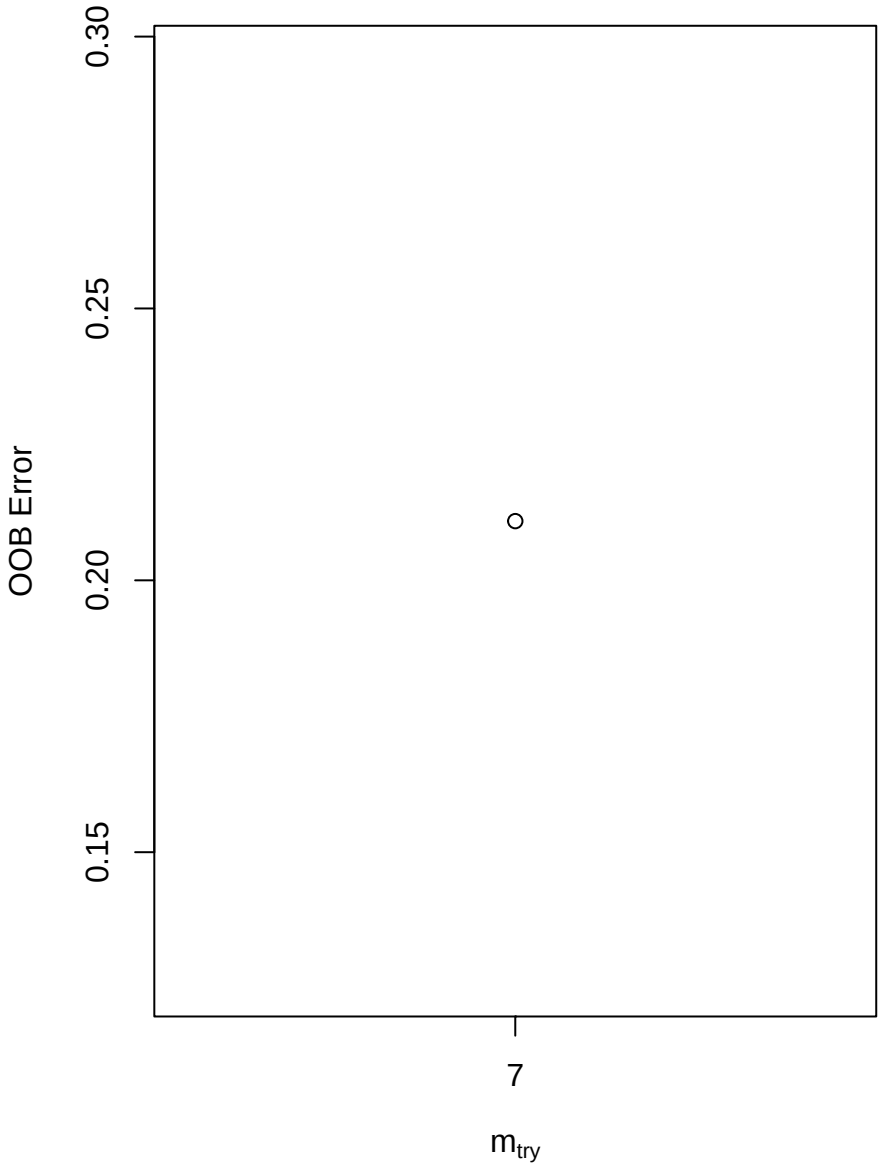
Regression Tree

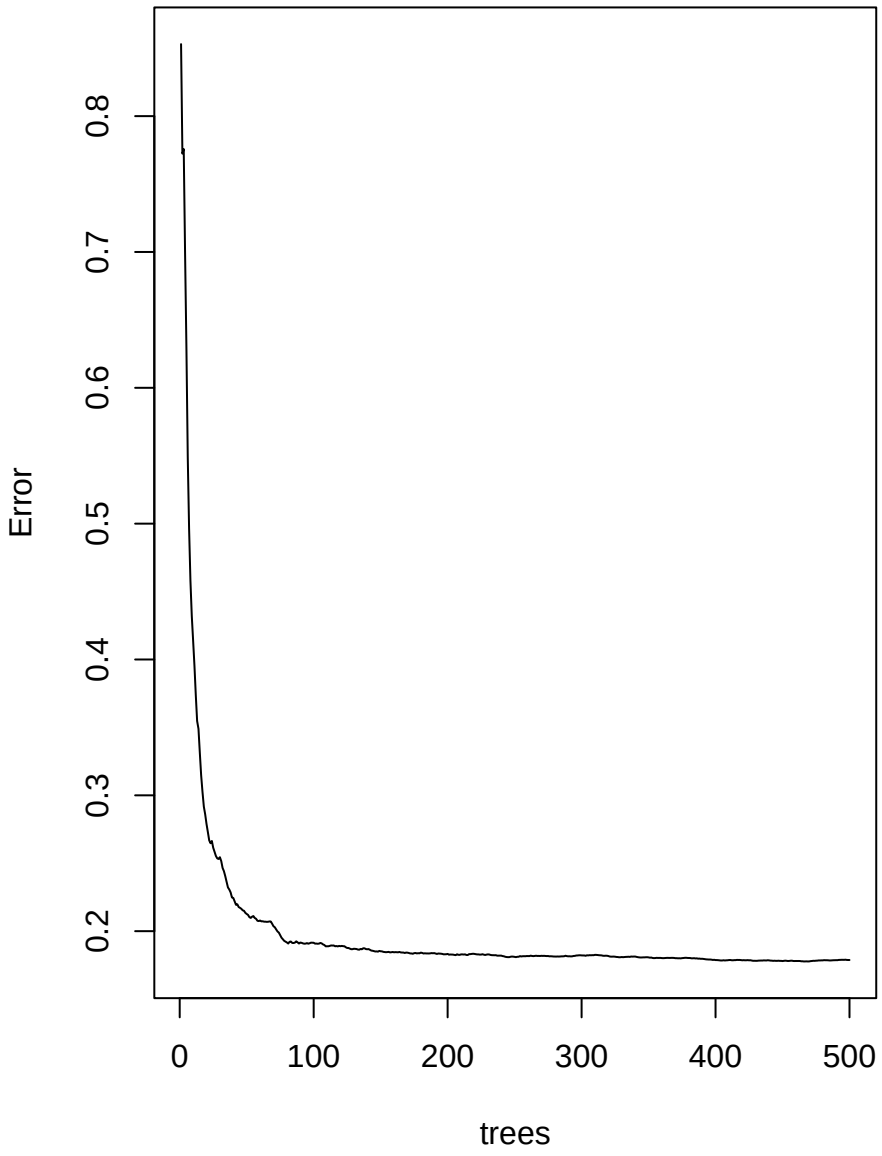


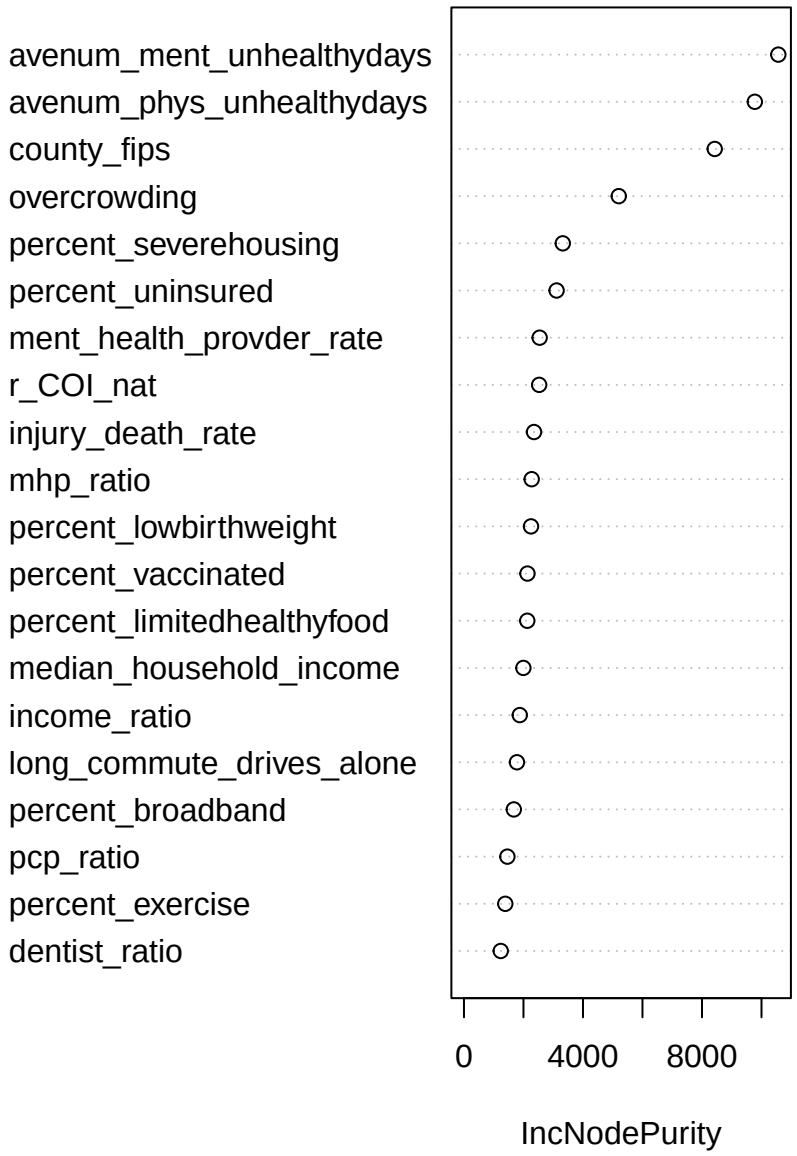
Random Forest

We will start by creating a random forest then identify the best mtry.


rf1
Error
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
trees


OOB Error
0.15
0.20
0.25
0.30
7
mtry


rf2
Error
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
100
200
300
400
500
trees



Comparison to Regression Tree

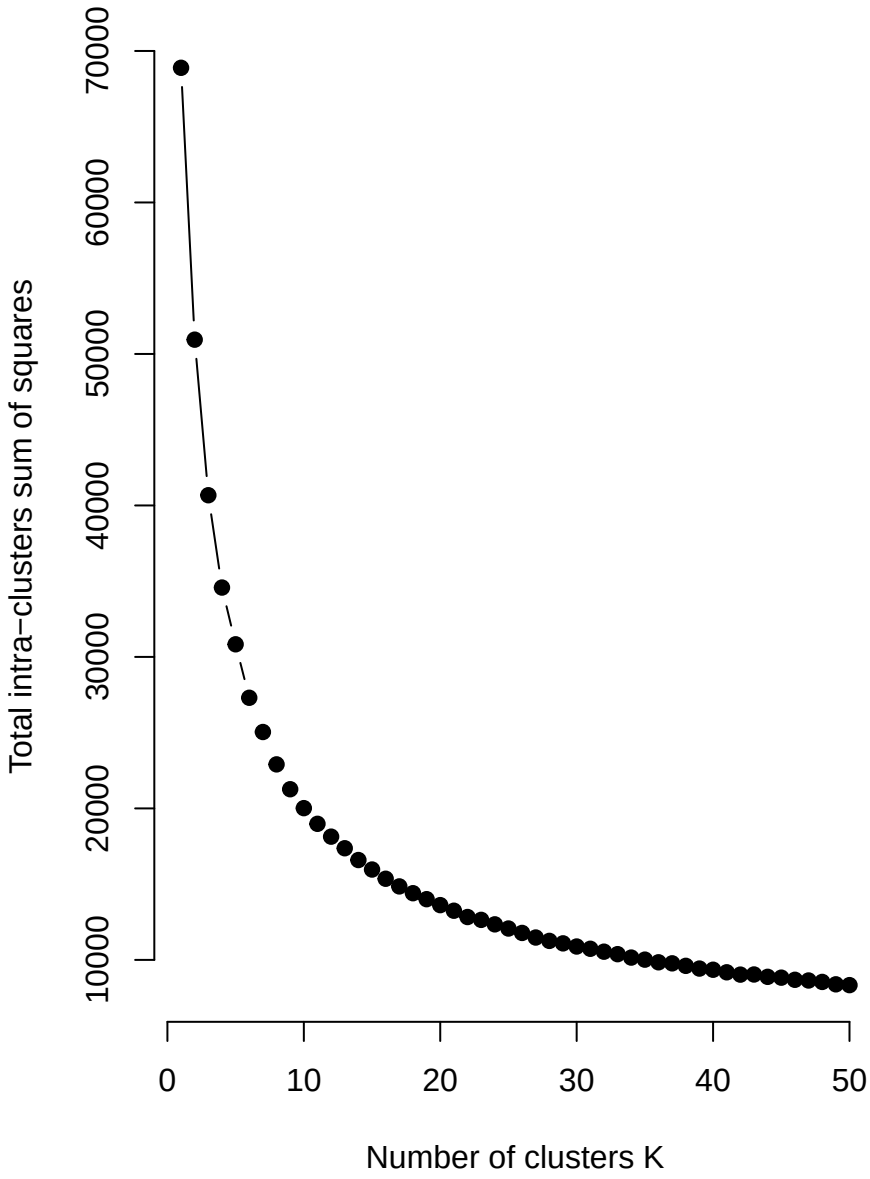
Artificial Neural Network

Comparing with the RF performance:

Unsupervised Learning

K-Means Clustering

```
WSS decreases at each step:

 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[37]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[49]  TRUE
```




```r
k_asym <- function(x,y,kd,kud){
  asymptote(
  x = x,
  y = y,
  degree = kd, # "optim",
  upper.degree = kud,
  # threshold = 0.90,  # Once y reaches 90% of asymptote
  proportional = TRUE,
  estimator = "mean", # "glm",
```

```
  ci.level = NULL # We don't need confidence intervals - prevent ERROR: object 'y_lwr' not found
  )
}

k_pop_asymptote_hunter <- k_asym(k.values, perc_variance, kpop_degree, kpop_uppper.degree)
```

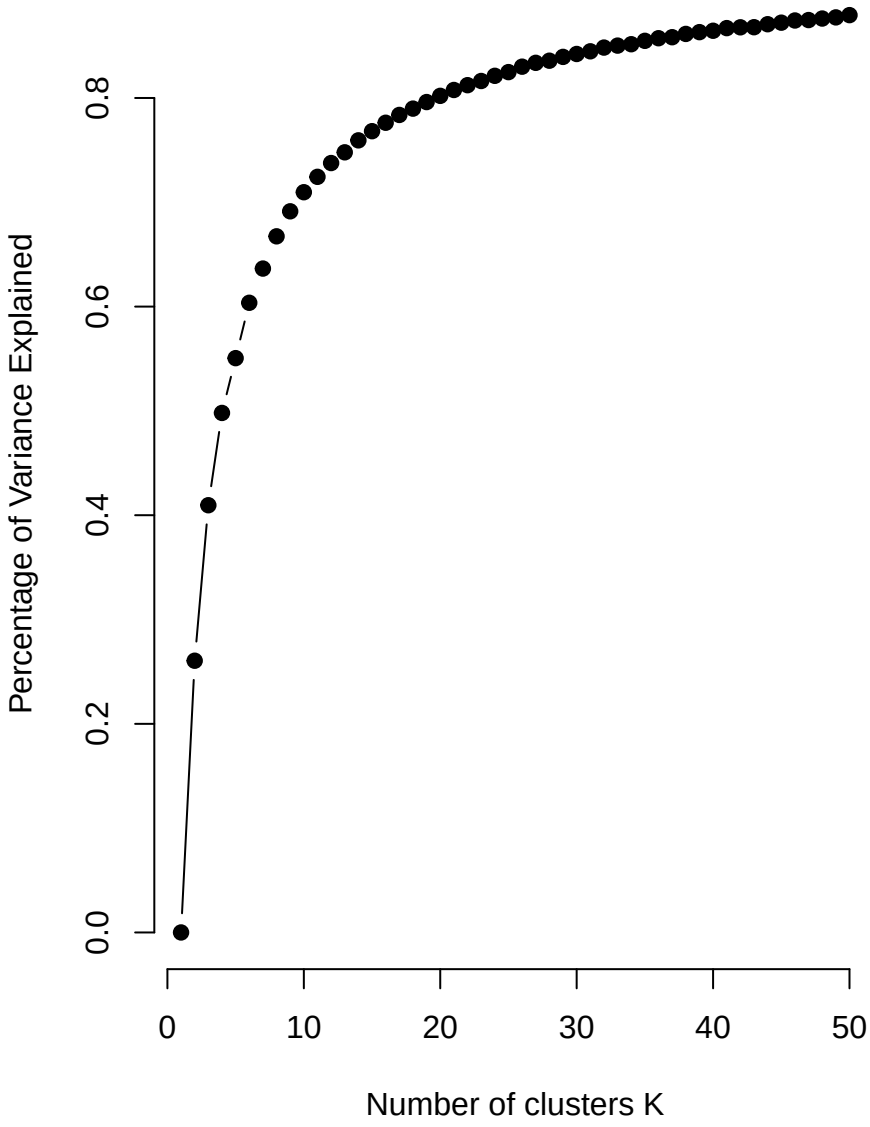
```
Estimated horizontal asymptote: ~0.9092164

Asymptote reached at x = 41
Asymptote threshold used: y = 0.8637556
Confidence level used:
```

```
k_pop_asymptote_hunter
```

```
$results
    x            y           ys
1   1 2.875478e-14 5.331310e-07
2   2 2.605167e-01 2.604983e-01
3   3 4.095796e-01 4.094271e-01
4   4 4.980493e-01 4.982111e-01
5   5 5.505960e-01 5.498405e-01
6   6 6.036484e-01 6.038241e-01
7   7 6.364631e-01 6.376481e-01
8   8 6.673735e-01 6.700458e-01
9   9 6.913457e-01 6.908645e-01
10 10 7.096535e-01 7.092055e-01
11 11 7.244000e-01 7.248836e-01
12 12 7.376142e-01 7.389473e-01
13 13 7.479071e-01 7.486680e-01
14 14 7.593835e-01 7.593167e-01
15 15 7.681582e-01 7.683787e-01
16 16 7.762562e-01 7.763851e-01
17 17 7.837342e-01 7.835146e-01
18 18 7.898059e-01 7.898320e-01
19 19 7.960728e-01 7.948223e-01
20 20 8.020408e-01 8.031387e-01
21 21 8.076634e-01 8.070868e-01
22 22 8.122757e-01 8.115342e-01
23 23 8.162806e-01 8.159233e-01
24 24 8.212451e-01 8.204610e-01
25 25 8.247851e-01 8.254002e-01
```

```
26 26 8.300172e-01 8.301812e-01
27 27 8.337019e-01 8.338310e-01
28 28 8.357314e-01 8.365984e-01
29 29 8.393837e-01 8.389803e-01
30 30 8.421749e-01 8.410062e-01
31 31 8.447330e-01 8.408901e-01
32 32 8.483132e-01 8.491822e-01
33 33 8.502156e-01 8.498613e-01
34 34 8.515314e-01 8.516346e-01
35 35 8.547859e-01 8.534589e-01
36 36 8.572833e-01 8.549687e-01
37 37 8.582486e-01 8.599888e-01
38 38 8.614104e-01 8.594049e-01
39 39 8.631075e-01 8.559542e-01
40 40 8.644352e-01 8.649997e-01
41 41 8.669753e-01 8.660349e-01
42 42 8.677334e-01 8.676358e-01
43 43 8.678531e-01 8.685422e-01
44 44 8.707150e-01 8.704369e-01
45 45 8.721936e-01 8.720754e-01
46 46 8.742822e-01 8.744581e-01
47 47 8.746892e-01 8.746380e-01
48 48 8.761594e-01 8.761644e-01
49 49 8.772457e-01 8.772498e-01
50 50 8.794261e-01 8.794260e-01

$h.asymptote
[1] 0.9092164

$min.n
[1] 41

$optimal.degree
[1] 2
```

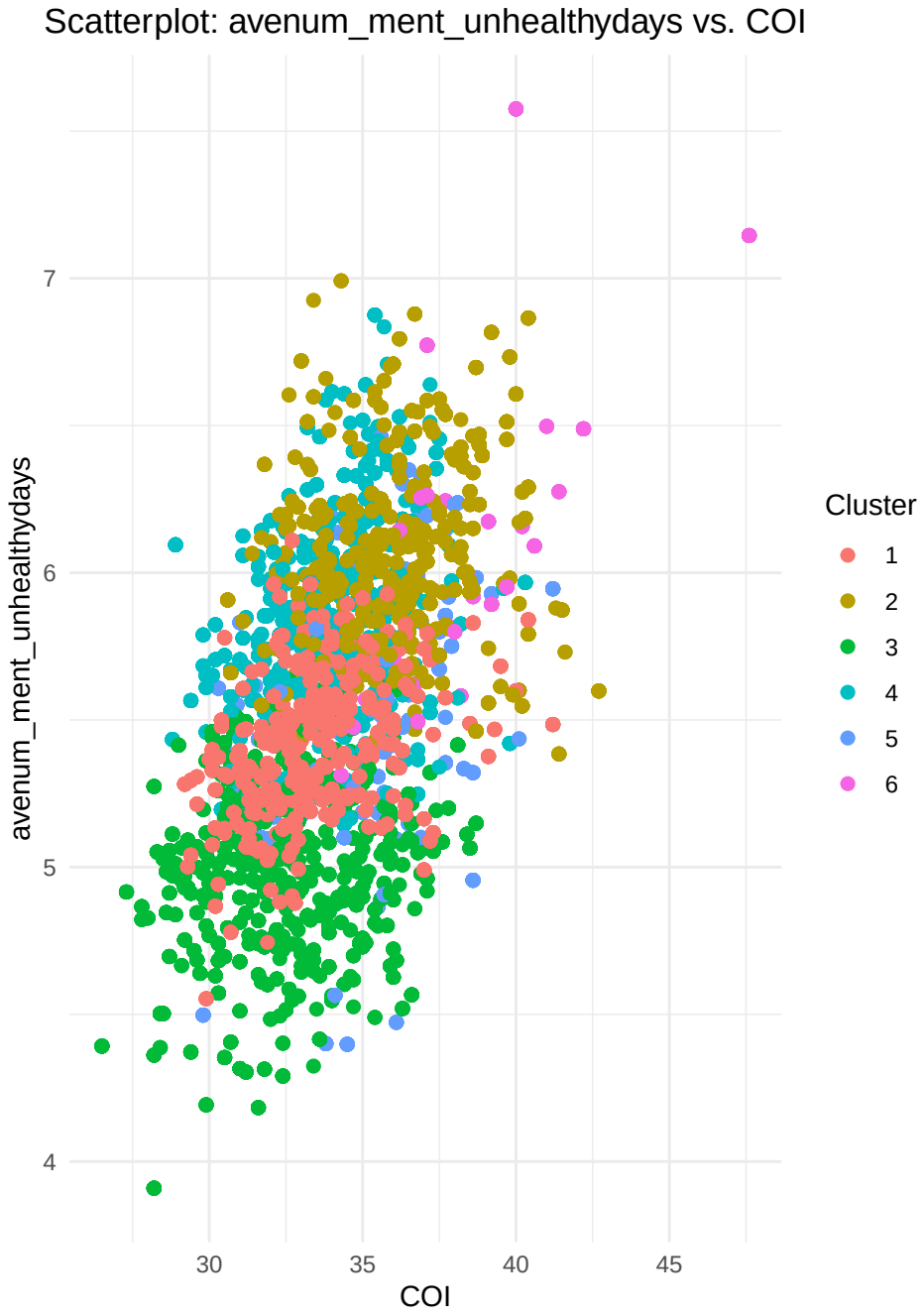
```r
plot(k.values, perc_variance,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Percentage of Variance Explained")
```



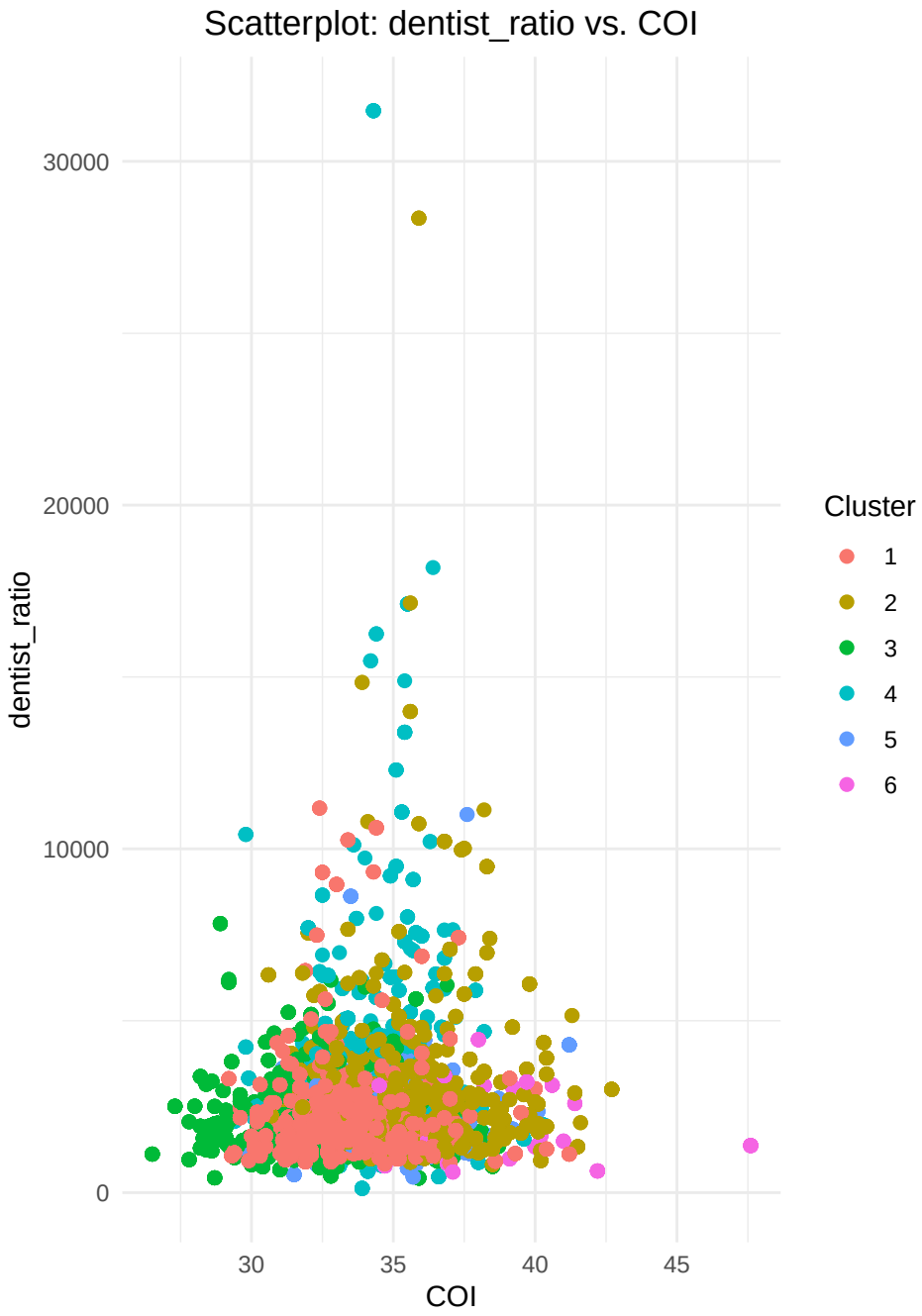

```
# points(k_pop_asymptote_hunter$min.n, k_pop_asymptote_hunter$h.asymptote, col = "blue", lwd = 2)

k_pop_best_K <- k_pop_asymptote_hunter$min.n
```

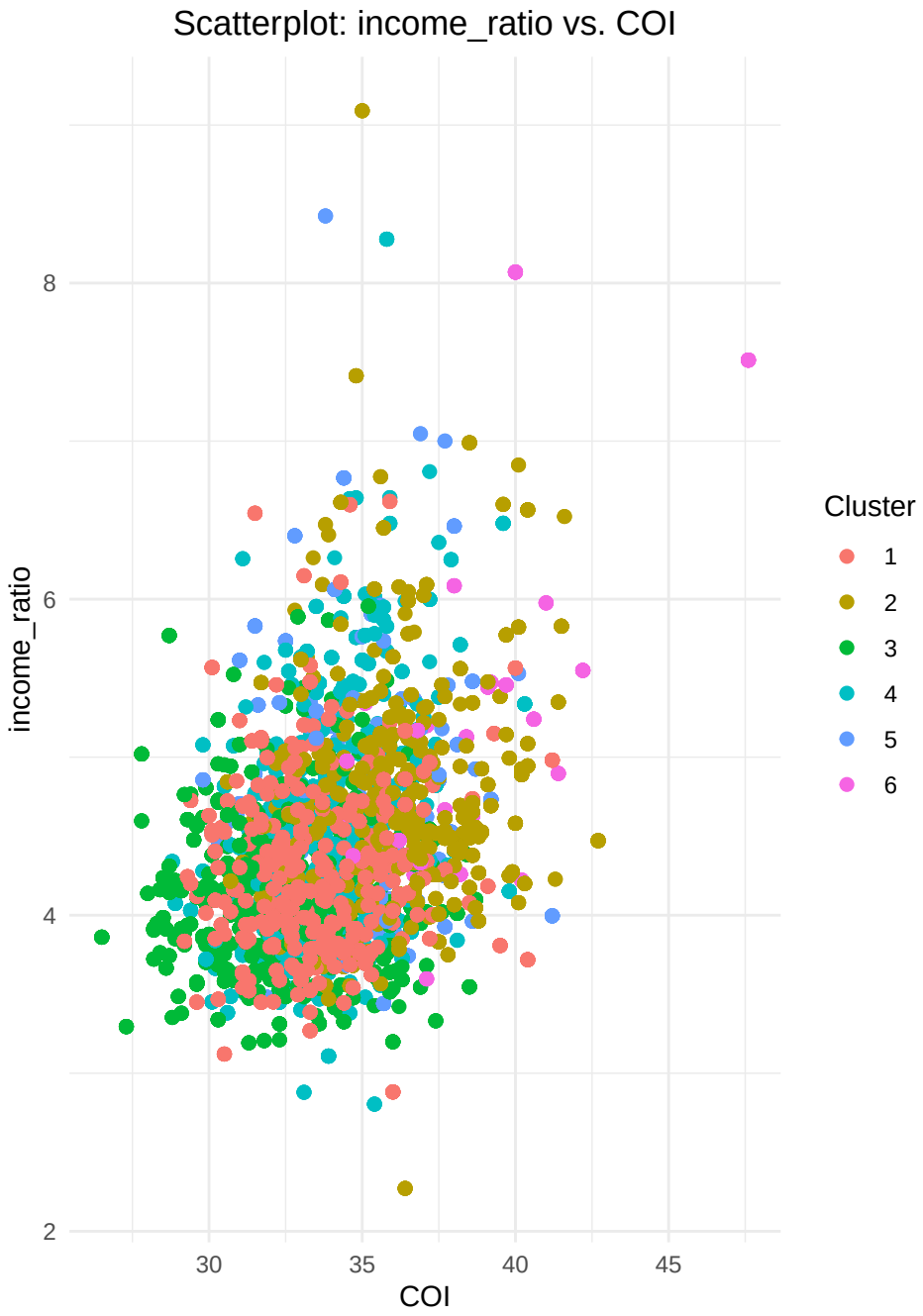
```
set.seed(seed_this)

## Kpop <- kmeans(df_cluster, k_pop_best_K, iter.max=100, nstart=50)
Kpop <- kmeans(df_cluster,6, iter.max=100, nstart=50)
df$ClusterNumber <- Kpop$cluster
```

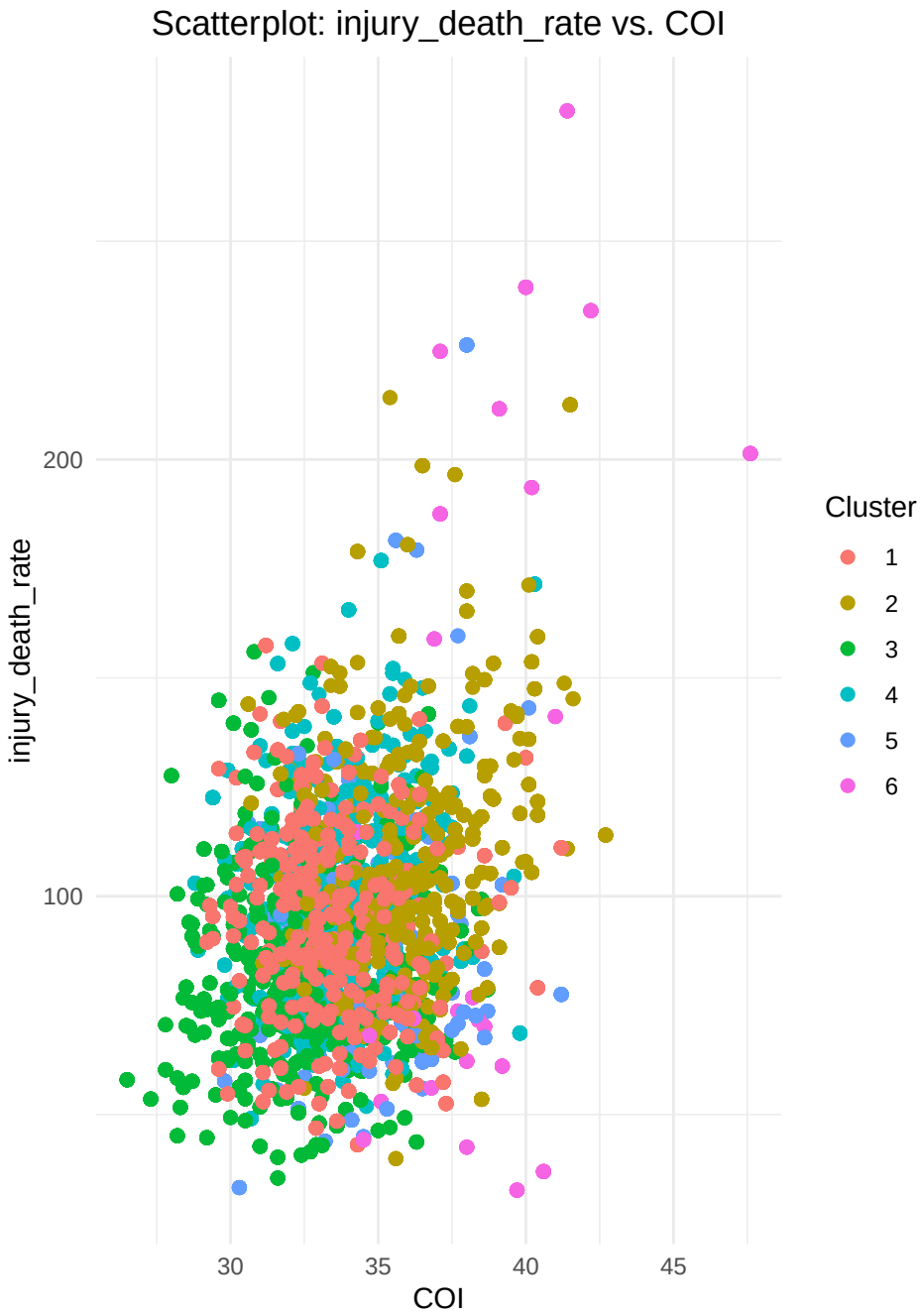
```
for(col in colsOfInterest) {
  # Use aes_string or .data pronoun for dynamic column selection
  p <- ggplot(df, aes(x=target, y=.data[[col]],
                      color=as.factor(ClusterNumber))) +
    geom_point(size=2) +
    labs(x = "COI", y = col, color = "Cluster") +
    ggtitle(paste("Scatterplot:", col, "vs. COI")) +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))

  print(p)

}
```


Scatterplot: avenum_ment_unhealthydays vs. COI
avenum_ment_unhealthydays
7
6
5
4
30
35
40
45
COI
Cluster
1
2
3
4
5
6

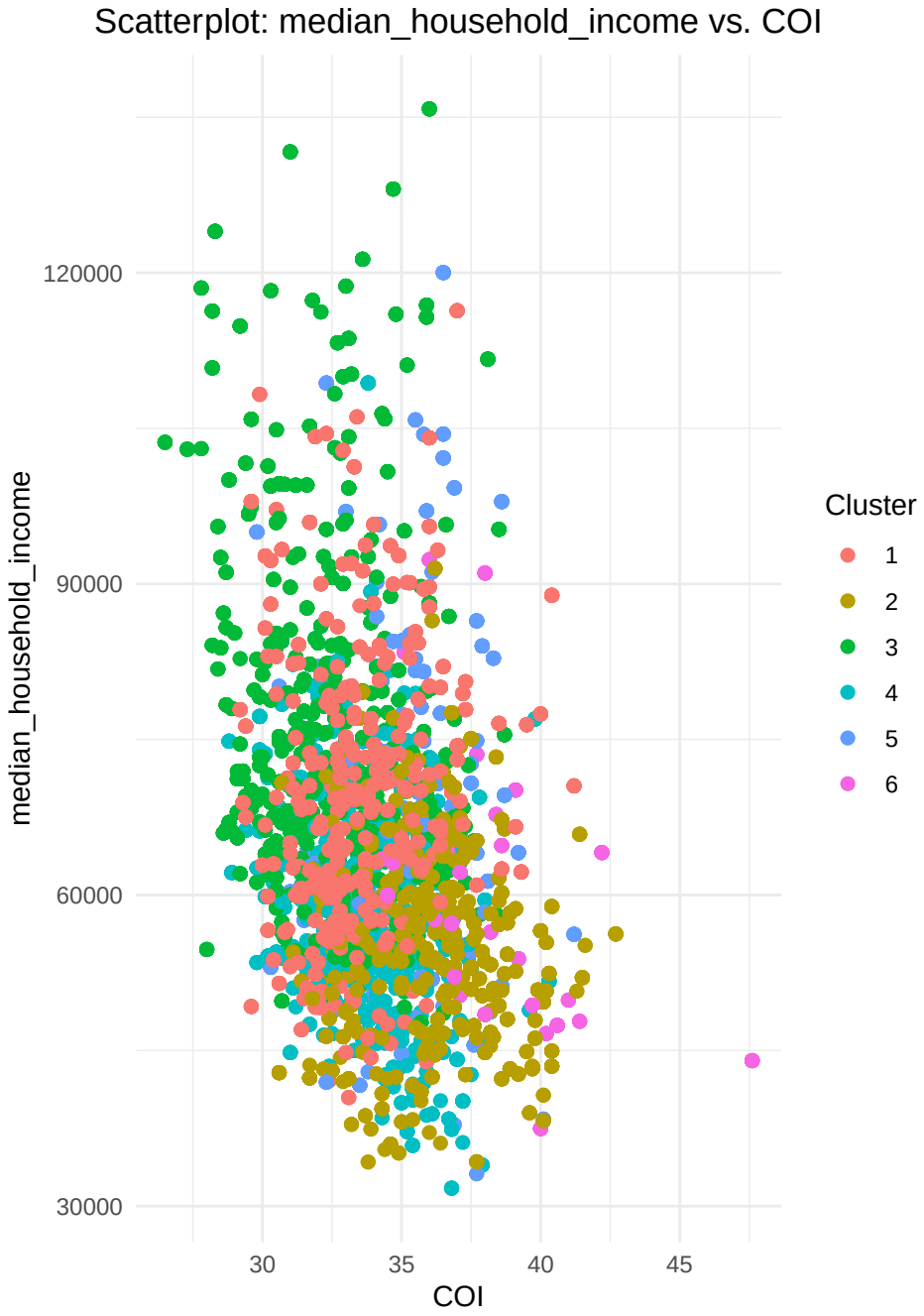

Scatterplot: avenum_phys_unhealthydays vs. COI
avenum_phys_unhealthydays
6
5
4
3
30
35
40
45
COI
Cluster
1
2
3
4
5
6

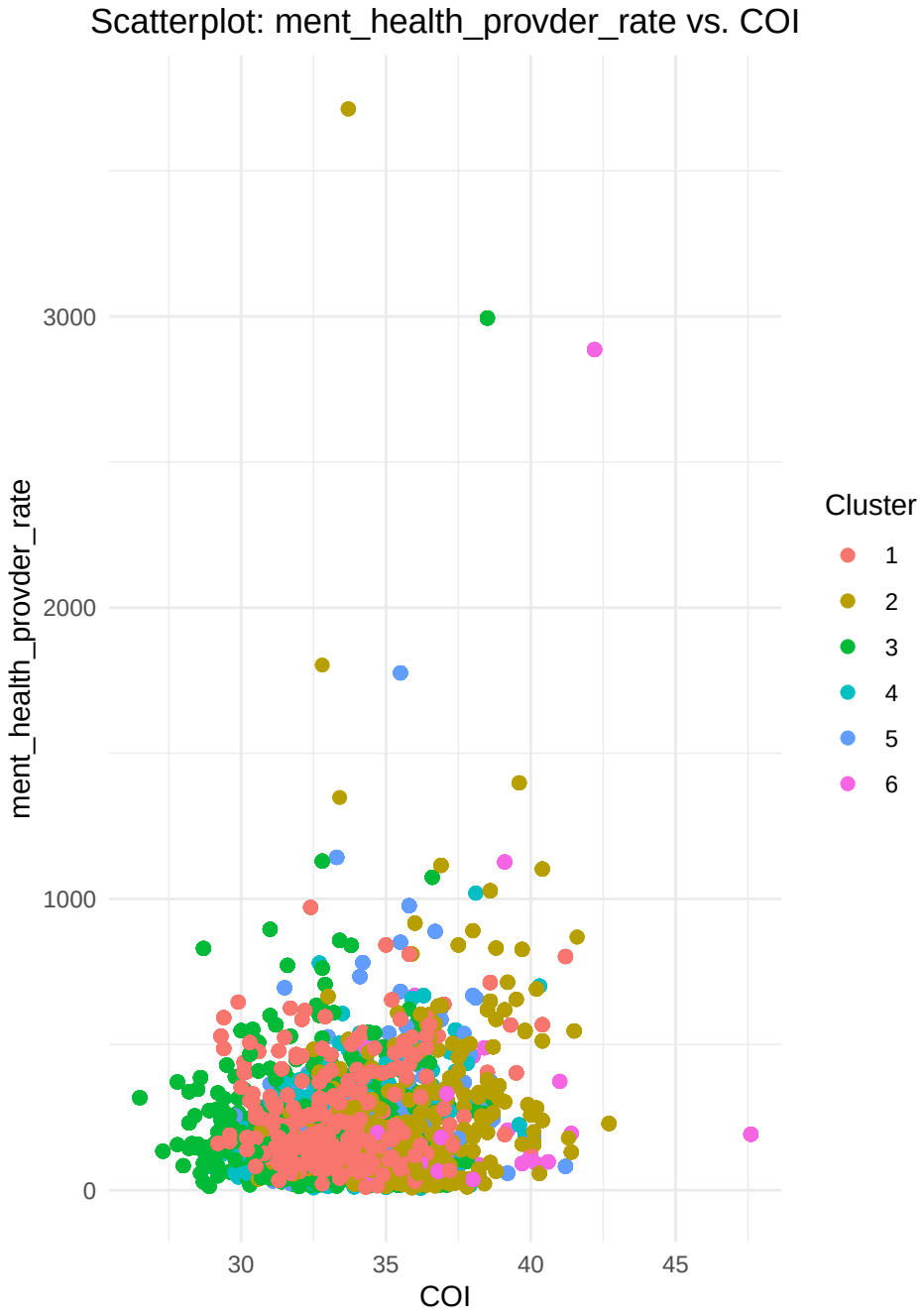

Scatterplot: dentist_ratio vs. COI
30000
20000
10000
0
dentist_ratio
30
35
40
45
COI
Cluster
1
2
3
4
5
6

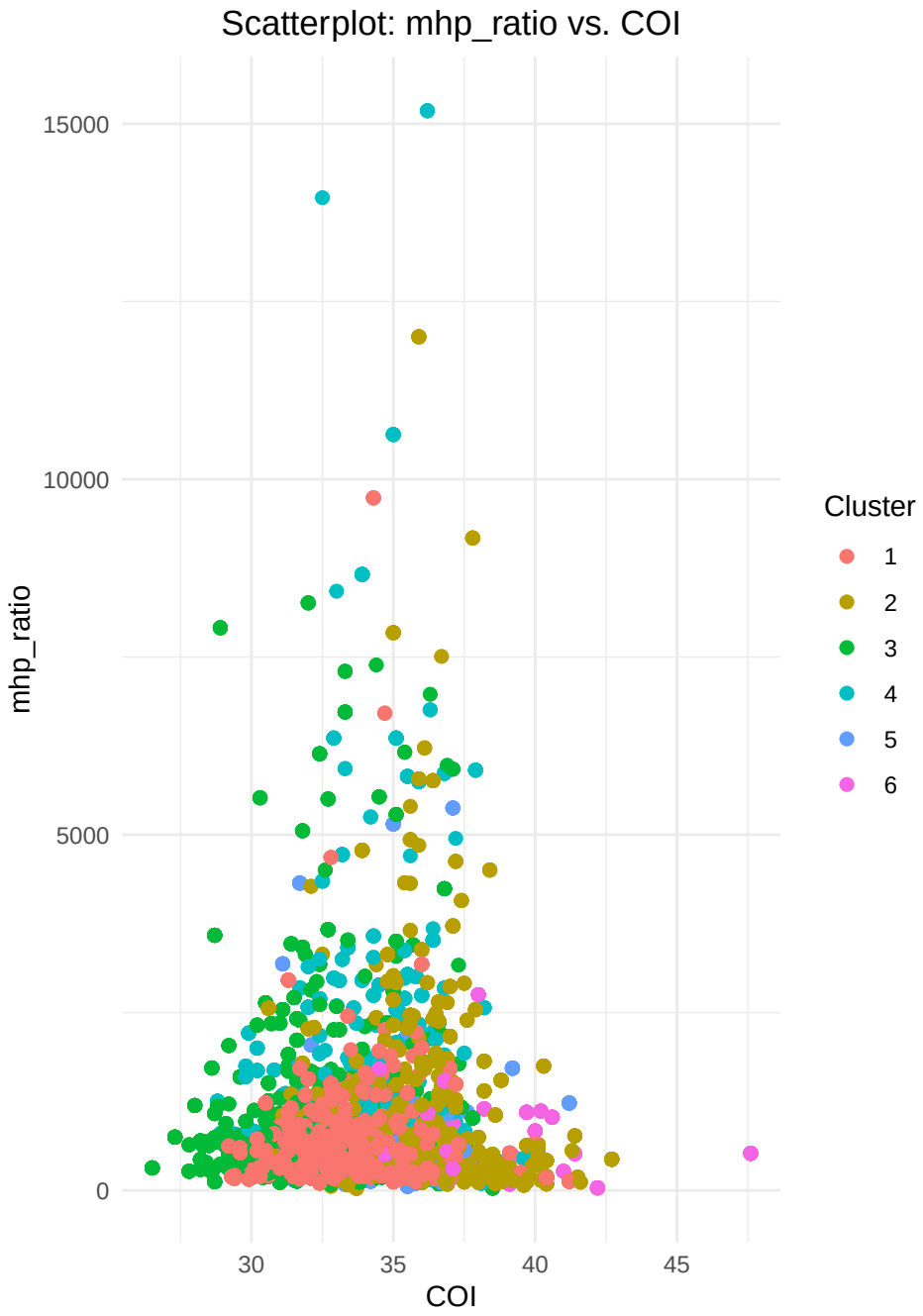

Scatterplot: income_ratio vs. COI
income_ratio
8
6
4
2
30
35
40
45
COI
Cluster
1
2
3
4
5
6

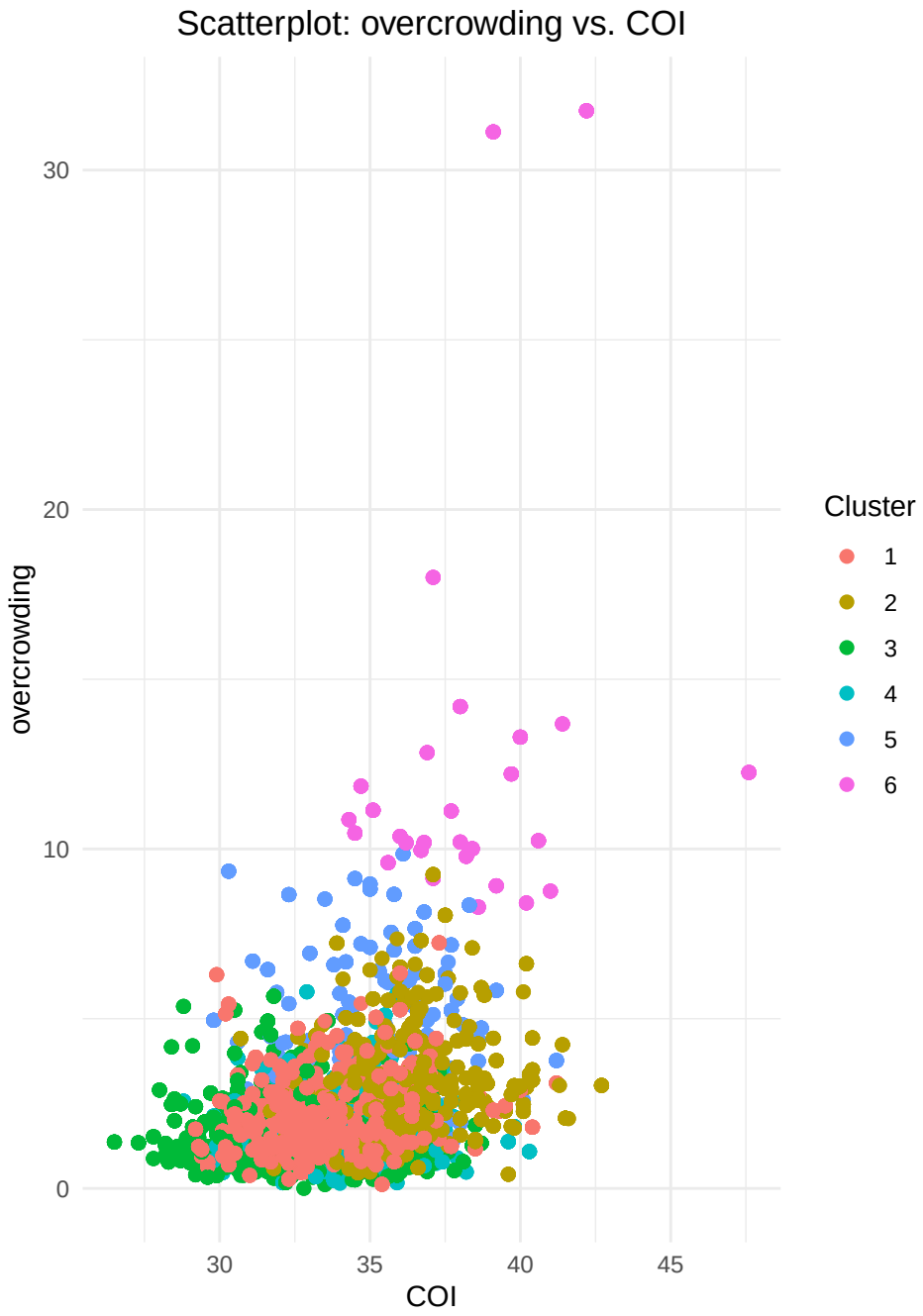

Scatterplot: injury_death_rate vs. COI
injury_death_rate
200
100
30
35
40
45
COI
Cluster
1
2
3
4
5
6

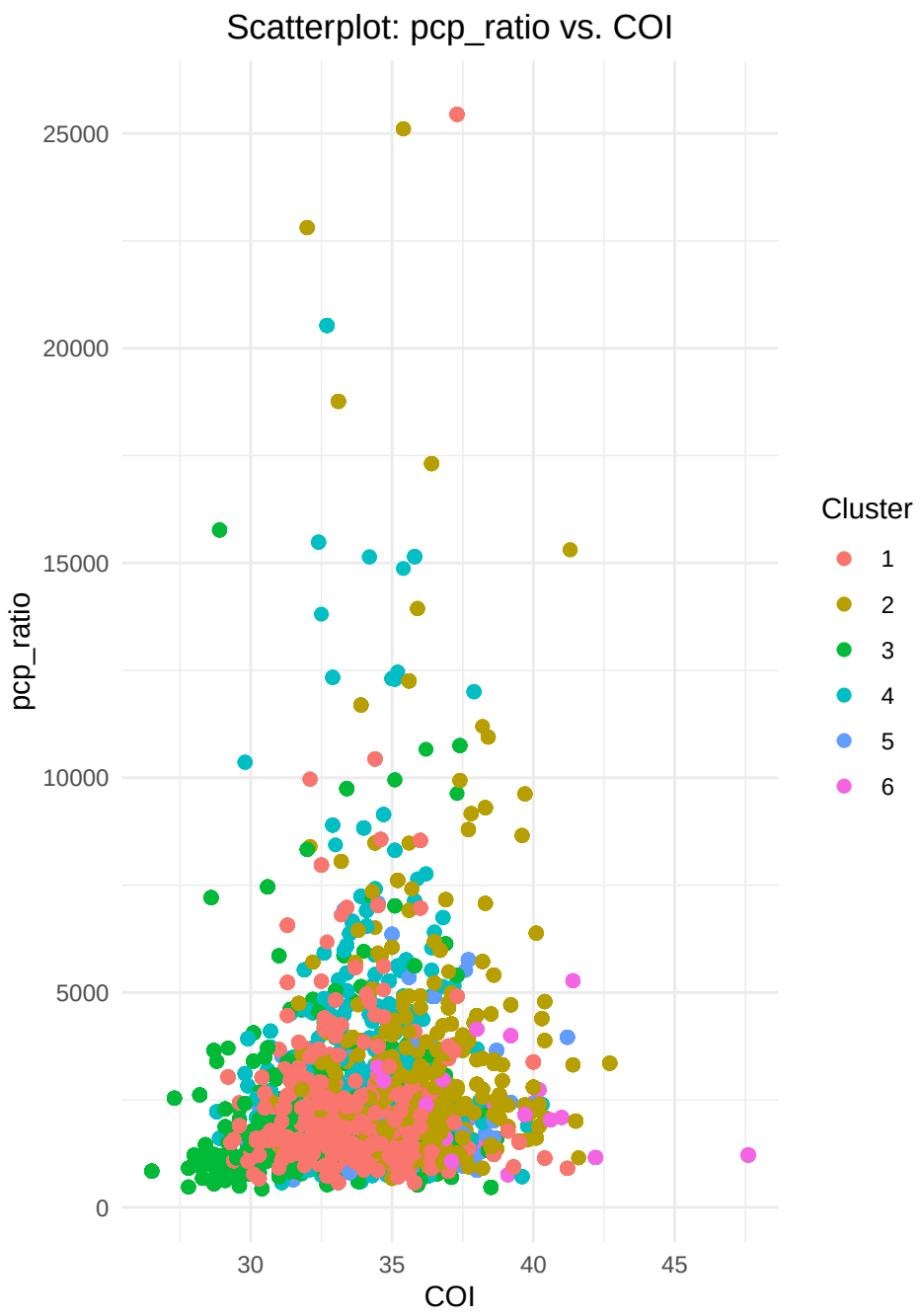

Scatterplot: long_commute_drives_alone vs. COI
long_commute_drives_alone
60
40
20
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6

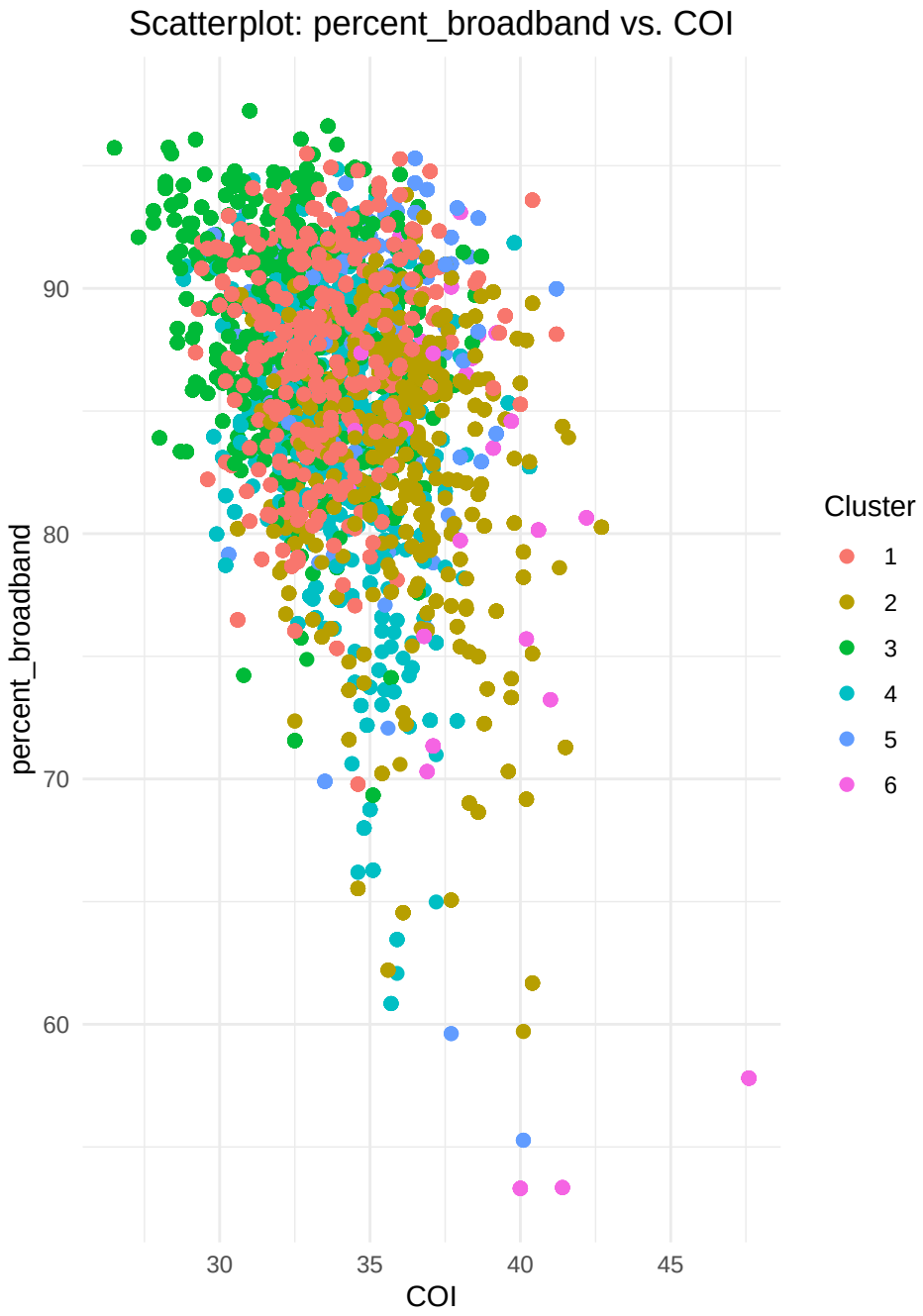

Scatterplot: median_household_income vs. COI
median_household_income
120000
90000
60000
30000
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: ment_health_provder_rate vs. COI
ment_health_provder_rate
3000
2000
1000
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6

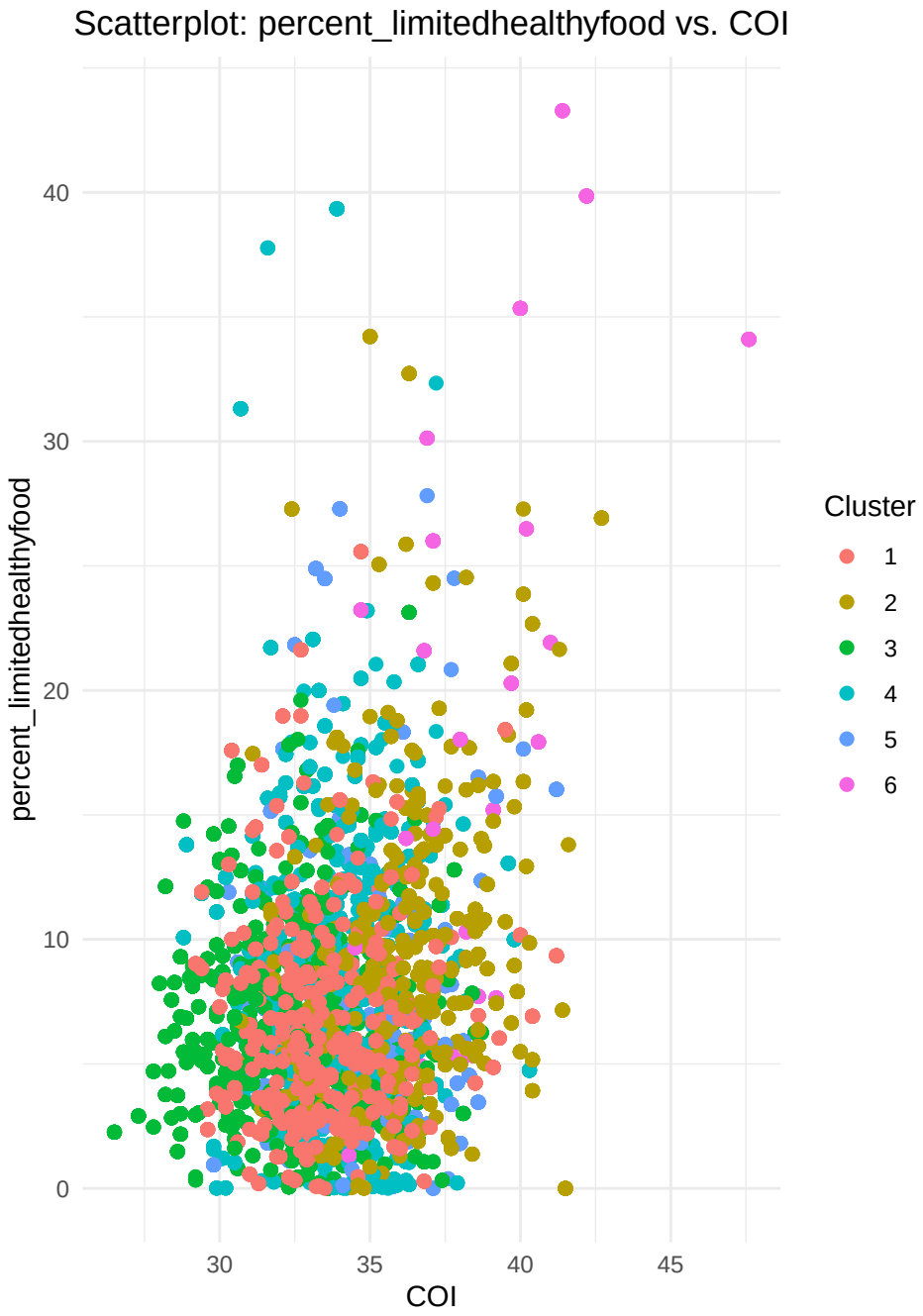

Scatterplot: mhp_ratio vs. COI
15000
10000
5000
0
mhp_ratio
30
35
40
45
COI
Cluster
1
2
3
4
5
6

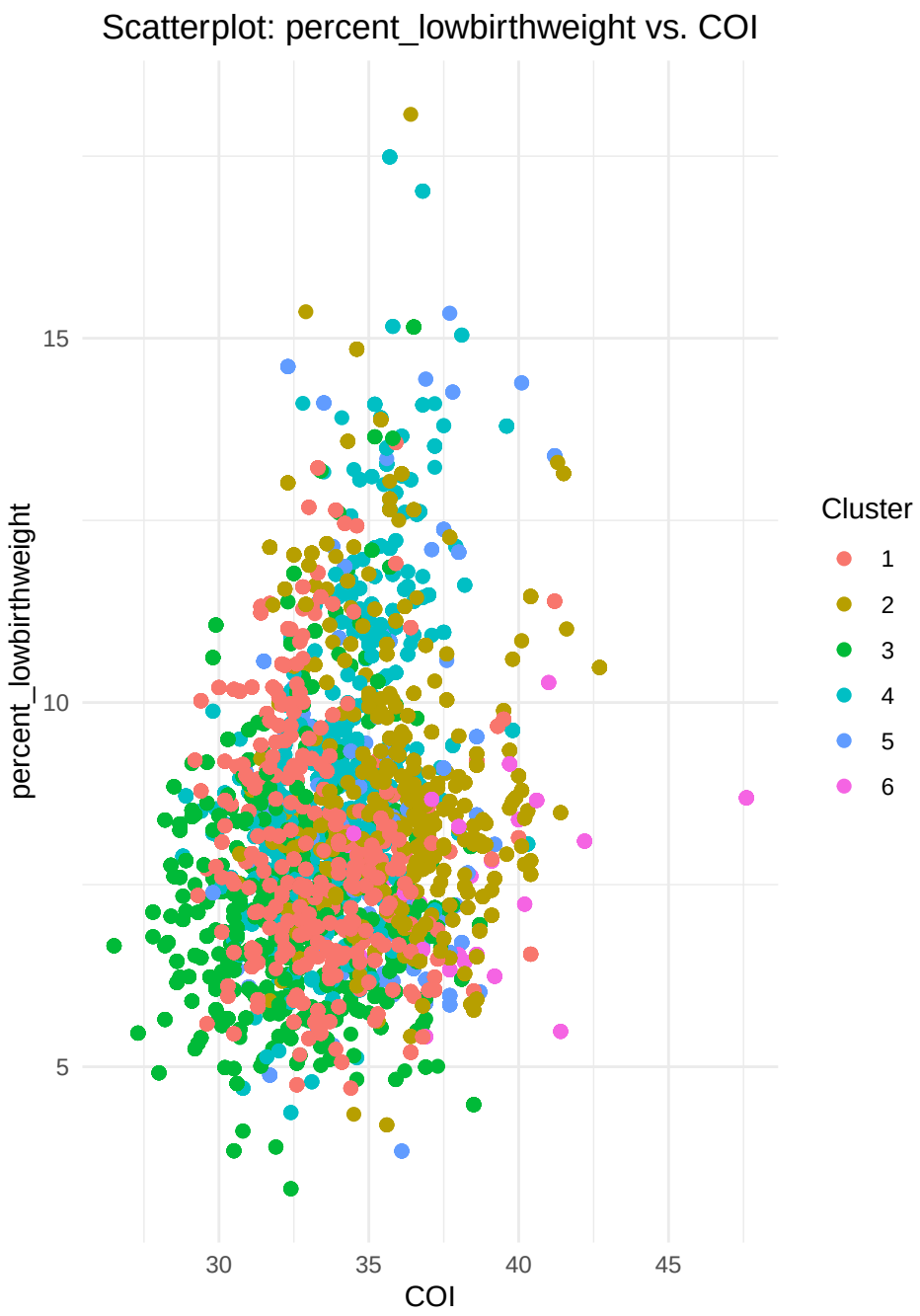

Scatterplot: overcrowding vs. COI
overcrowding
30
20
10
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6

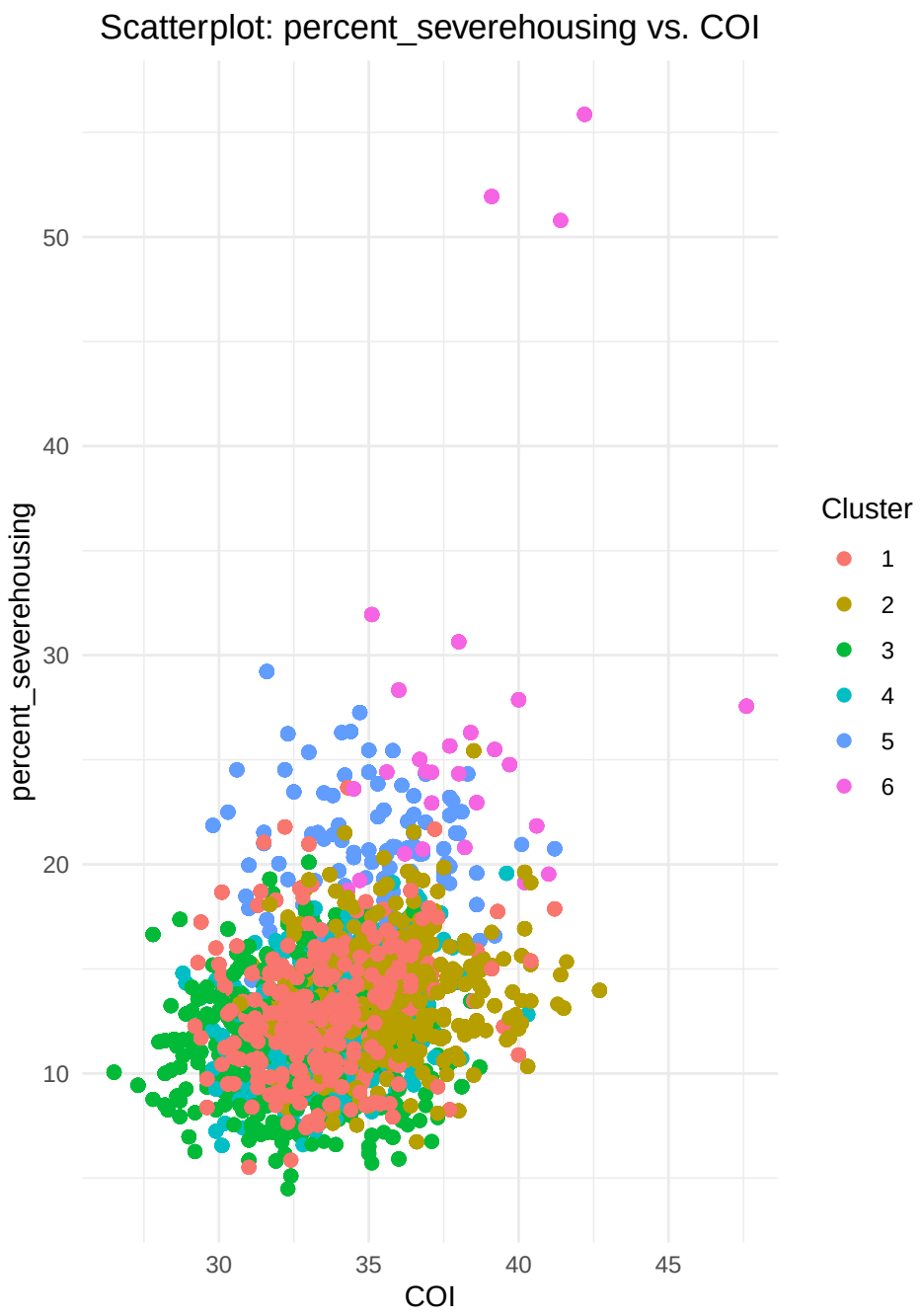

Scatterplot: pcp_ratio vs. COI
pcp_ratio
25000
20000
15000
10000
5000
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6

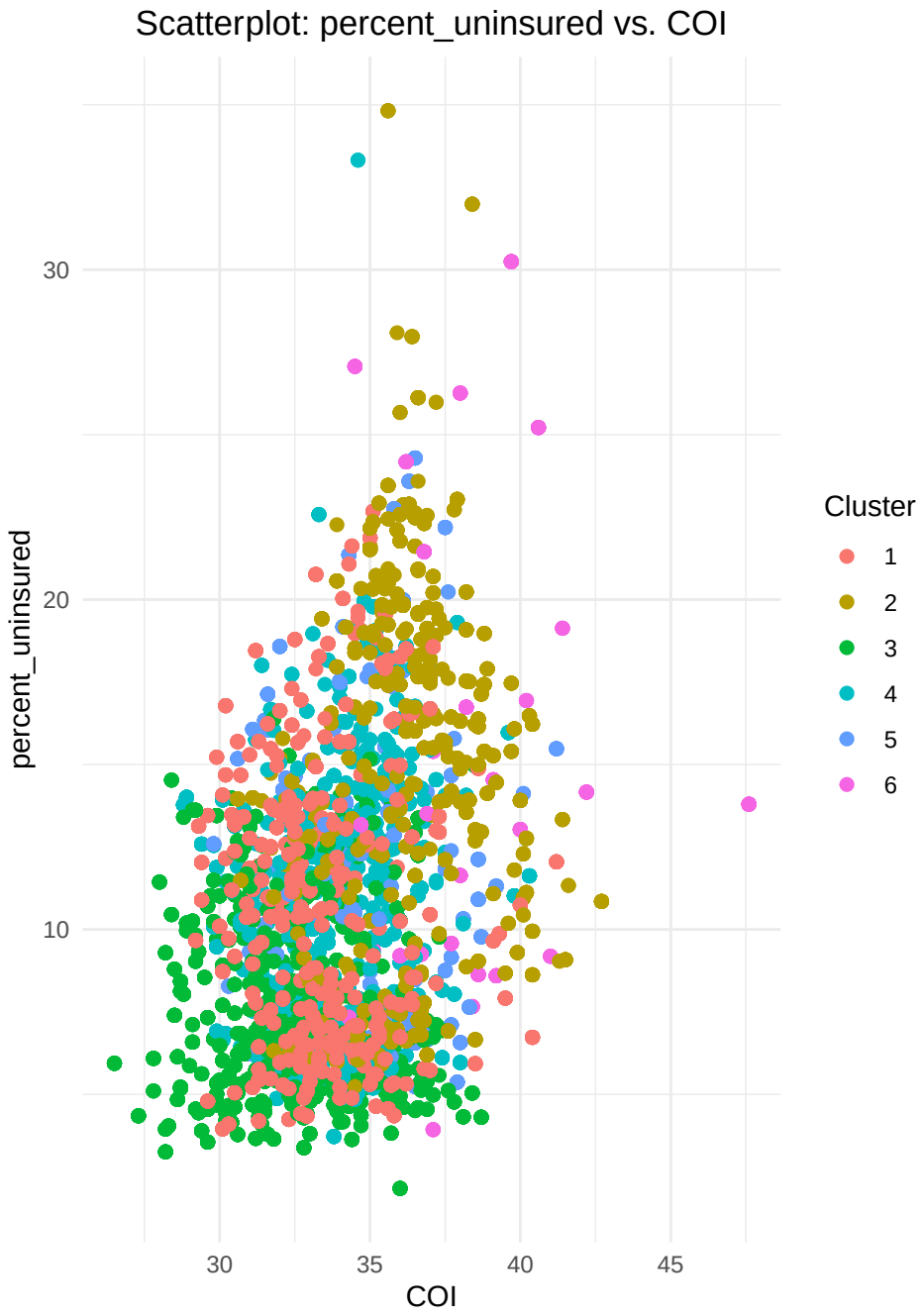

Scatterplot: percent_broadband vs. COI
percent_broadband
90
80
70
60
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: percent_exercise vs. COI
100
75
50
25
0
percent_exercise
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: percent_limitedhealthyfood vs. COI
percent_limitedhealthyfood
COI
0
10
20
30
40
30
35
40
45
Cluster
1
2
3
4
5
6


Scatterplot: percent_lowbirthweight vs. COI
percent_lowbirthweight
15
10
5
30
35
40
45
COI
Cluster
1
2
3
4
5
6

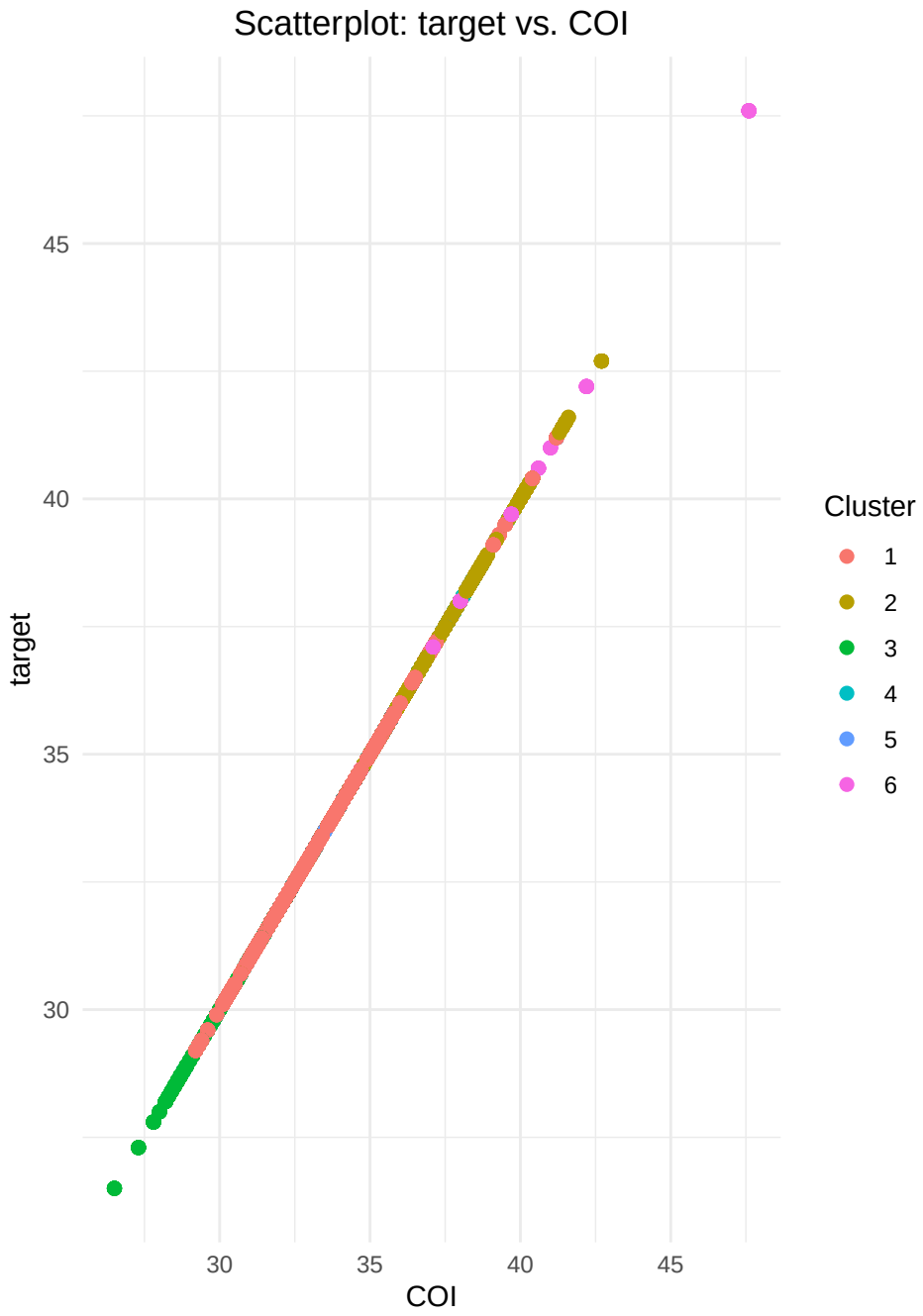

Scatterplot: percent_severehousing vs. COI
percent_severehousing
50
40
30
20
10
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: percent_uninsured vs. COI
percent_uninsured
30
20
10
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: percent_vaccinated vs. COI
percent_vaccinated
60
40
20
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: r_COI_nat vs. COI
r_COI_nat
75
50
25
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: county_fips vs. COI
county_fips
40000
20000
0
30
35
40
45
COI
Cluster
1
2
3
4
5
6


Scatterplot: target vs. COI
target
COI
45
40
35
30
30
35
40
45
Cluster
1
2
3
4
5
6

Compare

```
print(df_modelCompare)
```

```
                         Model     OSR2_iteration_01 OSR2_iteration_02
1                  Target Value percent_feelinglonely                NA
2                          Seed                 21813                NA
3              Linear Regression                  <NA>                NA
4                Regression Tree     0.620719160829882                NA
5                  Random Forest     0.972006982989319                NA
6  Artificial Neural Network ONE     0.435072725390803                NA
7 Artificial Neural Network ZERO     0.399148964987301                NA
  OSR2_iteration_03             OSR2_this
1                NA percent_feelinglonely
2                NA                 14273
3                NA                  <NA>
4                NA     0.576182452218744
5                NA     0.974455560966907
6                NA     0.414049201399109
7                NA     0.377337172524094
```

References

https://www.diversitydatakids.org/research-library/child-opportunity-index-30-2023-county-data

https://www.diversitydatakids.org/research-library/research-brief/what-child-opportunity

https://www.ers.usda.gov/data-products/urban-influence-codes