VAISHNAVI GANNAVARAM FINAL PROJECT MS ANALYTICS

```
#Loading the data set
library(readr)
Medicalcosts <- read_csv("med-insurance-cost.csv")
```

```
## Rows: 1338 Columns: 8
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): sex, smoker, region
## dbl (5): age, bmi, children, charges, Marital stat
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#View(Medicalcosts)

#copying the dataframe
medcostdf <- Medicalcosts

summary(medcostdf)
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.13   Max.   :5.000
##     smoker             region             charges       Marital stat
##  Length:1338        Length:1338        Min.   : 1122   Min.   :0.0000
##  Class :character   Class :character   1st Qu.: 4740   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median : 9382   Median :1.0000
##                                        Mean   :13270   Mean   :0.5531
##                                        3rd Qu.:16640   3rd Qu.:1.0000
##                                        Max.   :63770   Max.   :1.0000
```

```
#removings missing values
medcostdf<-na.omit(medcostdf)
summary(medcostdf)
```

```
##       age            sex                 bmi           children
##  Min.   :18.00   Length:1338        Min.   :15.96   Min.   :0.000
##  1st Qu.:27.00   Class :character   1st Qu.:26.30   1st Qu.:0.000
##  Median :39.00   Mode  :character   Median :30.40   Median :1.000
##  Mean   :39.21                      Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                      3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                      Max.   :53.13   Max.   :5.000
##     smoker             region             charges       Marital stat
##  Length:1338        Length:1338        Min.   : 1122   Min.   :0.0000
##  Class :character   Class :character   1st Qu.: 4740   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median : 9382   Median :1.0000
##                                        Mean   :13270   Mean   :0.5531
##                                        3rd Qu.:16640   3rd Qu.:1.0000
##                                        Max.   :63770   Max.   :1.0000
```

```
###RECODING CATEGORICAL VARIABLES TO NUMERICS
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(forcats)

medcostnewdf <- medcostdf %>%
                  mutate(sex = fct_recode(sex,
                                          "0" = "female",
                                          "1"= "male")) %>%
                     mutate(smoker = fct_recode(smoker,
                                          "1"="yes",
                                          "0"="no"))  %>%
                       mutate(region = fct_recode(region,
                                          "1"= "southwest",
                                          "2"= "southeast",
                                          "3"= "northwest",
                                          "4"= "northeast"))
medcostnewdf
```

```
## # A tibble: 1,338 × 8
##      age sex     bmi children smoker region charges `Marital stat`
##    <dbl> <fct> <dbl>    <dbl> <fct>  <fct>    <dbl>          <dbl>
##  1    19 0      27.9        0 1      1       16885.              0
##  2    18 1      33.8        1 0      2        1726.              1
##  3    28 1      33          3 0      2        4449.              1
##  4    33 1      22.7        0 0      3       21984.              1
##  5    32 1      28.9        0 0      3        3867.              1
##  6    31 0      25.7        0 0      2        3757.              0
##  7    46 0      33.4        1 0      2        8241.              0
##  8    37 0      27.7        3 0      3        7282.              1
##  9    37 1      29.8        2 0      4        6406.              0
## 10    60 0      25.8        0 0      3       28923.              1
## # ℹ 1,328 more rows
```

DATA VISUALIZATION

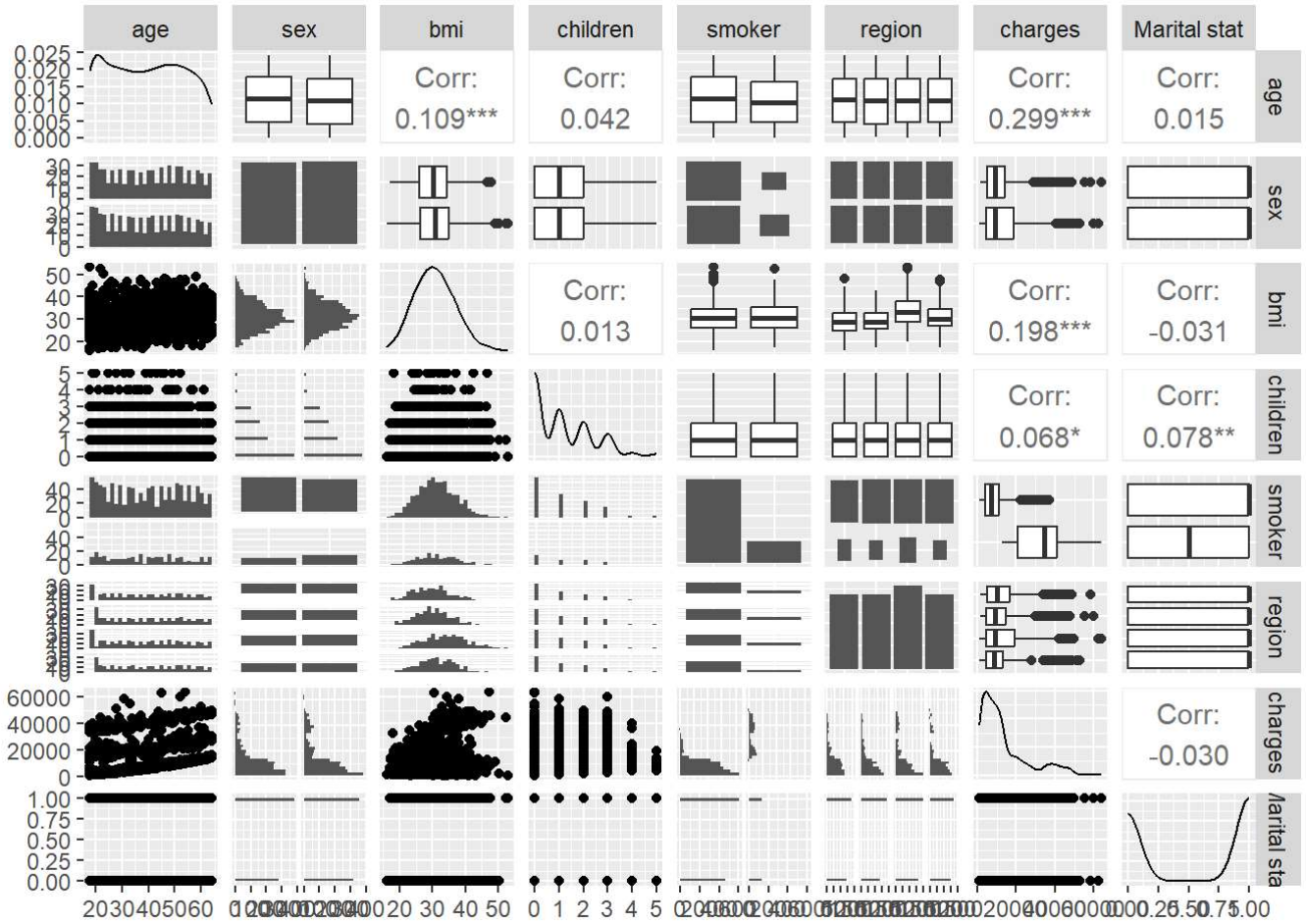
```
#plotting ggpairs to look at the data in each variable individually and appropriate pairs (predi
ctor + outcome combinations) and note anything “interesting”.

library("ggplot2")
library("GGally")
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(medcostnewdf)
```

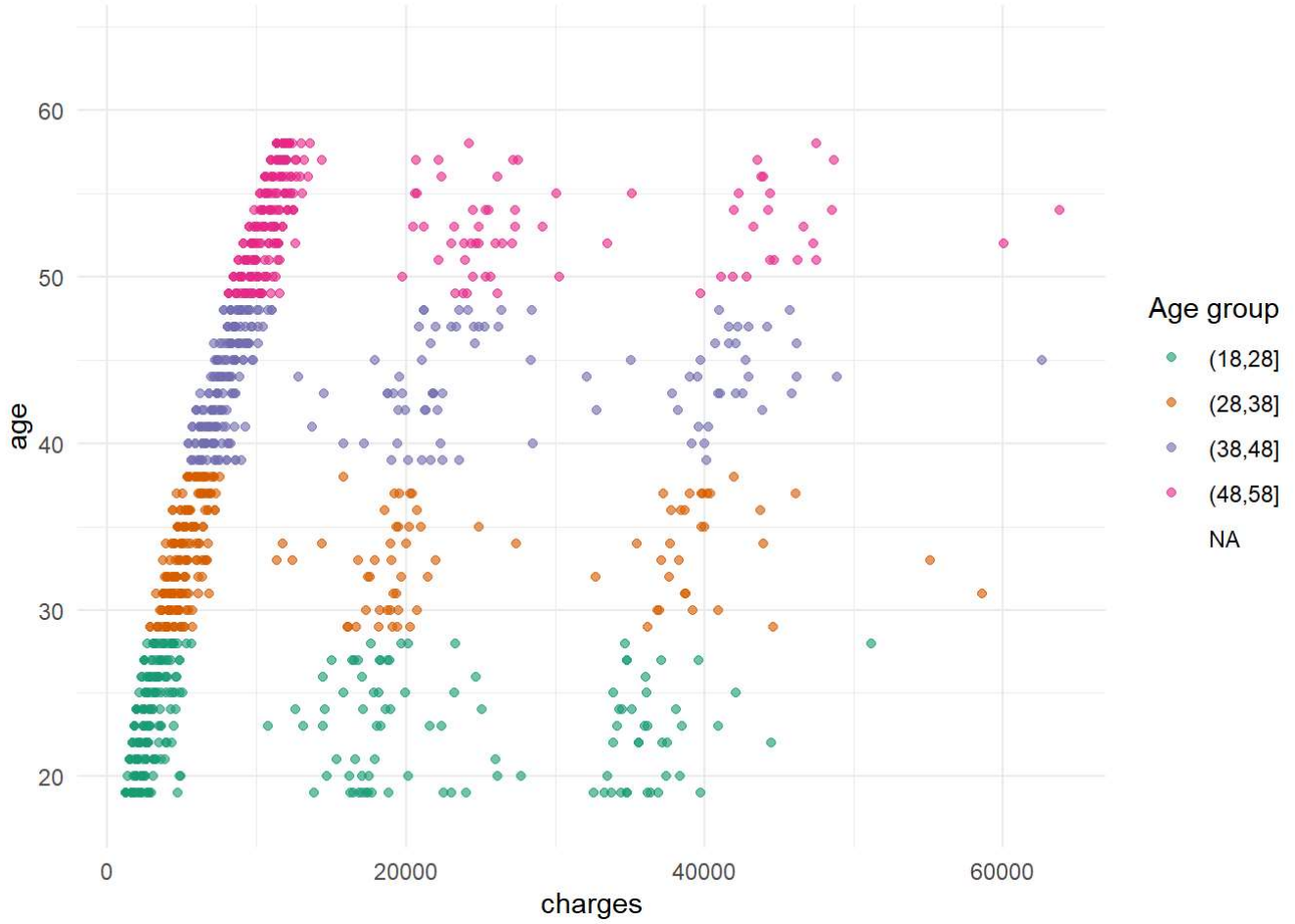
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
library(ggplot2)
library(dplyr)

agecharges <- medcostnewdf %>%
  mutate(age_group = cut(age, breaks = seq(18, 64, by = 10))) %>%
  ggplot(aes(x = charges, y = age, color = age_group)) +
  geom_point(alpha = .6) +
  labs(x = "charges", y = "age", color = "Age group") +
  scale_color_brewer(palette = "Dark2") +
  theme_minimal()
agecharges
```

```
## Warning: Removed 208 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



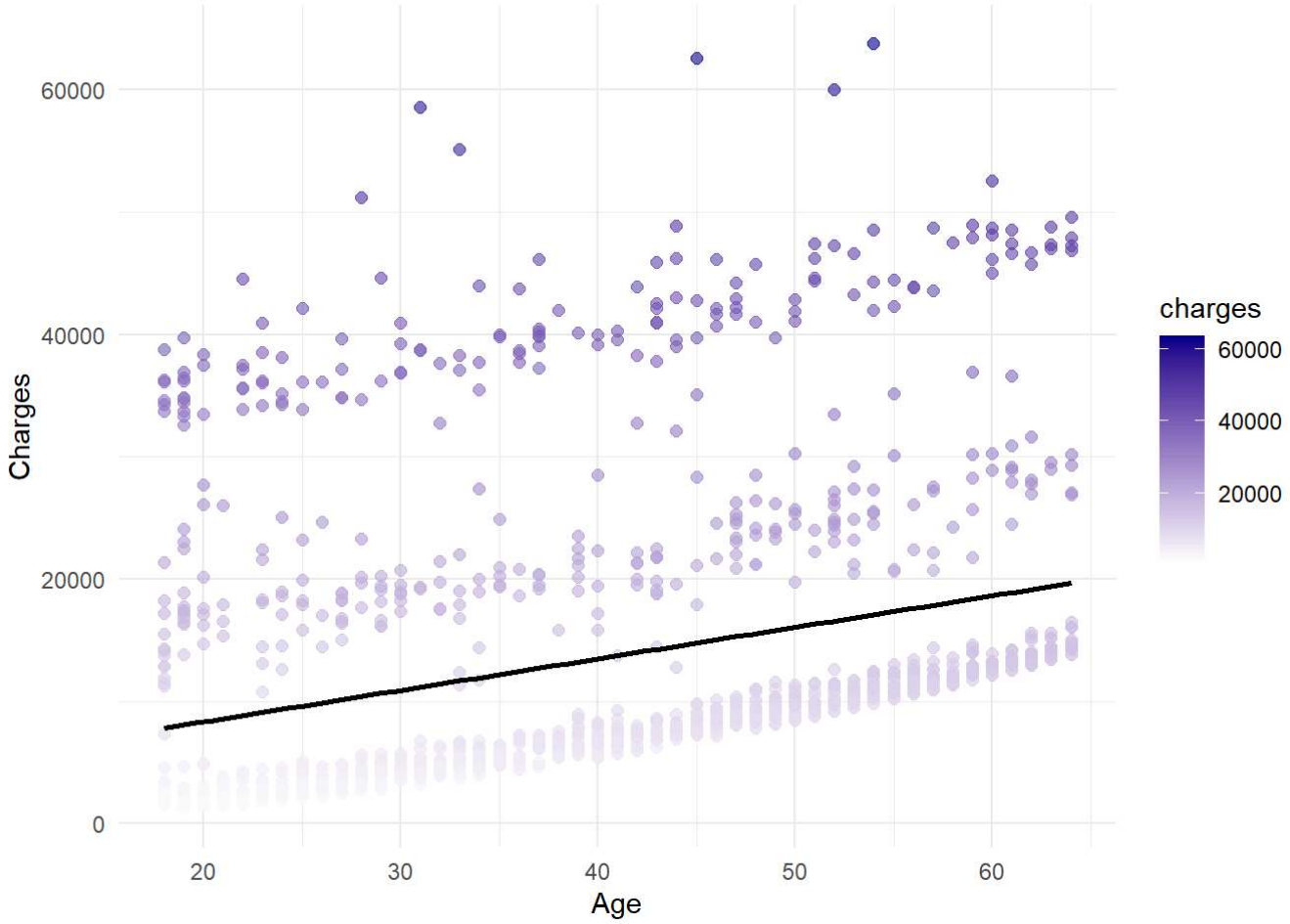

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

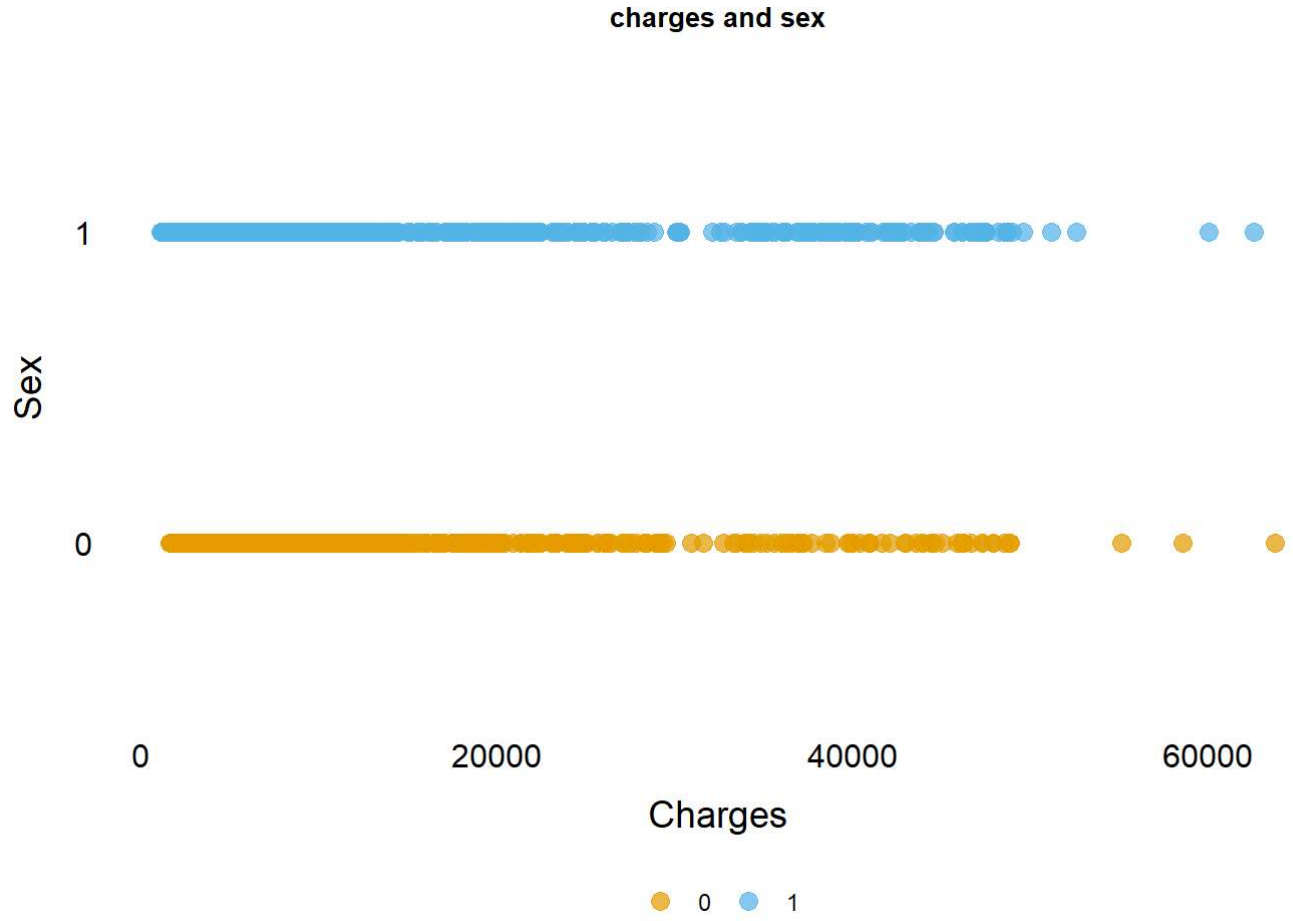
```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
charges_and_age <- medcostnewdf %>%
  ggplot(aes(x = age, y = charges, color = charges)) +
  geom_point(size = 2, alpha = 0.6) +
  scale_color_gradient(low = "white", high = "darkblue") +
  stat_smooth(method = "lm", se = FALSE, aes(group = 1), color = "black") +
  labs(x = "Age", y = "Charges") +
  theme_minimal()

charges_and_age
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


60000
40000
20000
0
Charges
20
30
40
50
60
Age
charges
60000
40000
20000

```
library(ggplot2)

# define custom colors
my_colors <- c("#E69F00", "#56B4E9")

# create scatter plot with custom colors
charges_and_sex <- ggplot(medcostnewdf, aes(x = charges, y = sex, color = sex)) +
  geom_point(alpha = .7, size = 3) +
  scale_color_manual(values = my_colors) +
  labs(x = "Charges", y = "Sex") +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    legend.title = element_blank(),
    legend.direction = "horizontal",
    legend.box.just = "center",
    axis.text.x = element_text(size = 12, color = "black"),
    axis.text.y = element_text(size = 12, color = "black"),
    axis.title.x = element_text(size = 14, color = "black", margin = margin(t = 10)),
    axis.title.y = element_text(size = 14, color = "black", margin = margin(r = 10)),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    plot.title = element_text(size = 10, hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(size = 16, hjust = 0.5),
    plot.caption = element_text(size = 12, hjust = 0, color = "grey50", margin = margin(t = 10,
b = 5))
  ) +
  ggtitle("charges and sex")


charges_and_sex
```



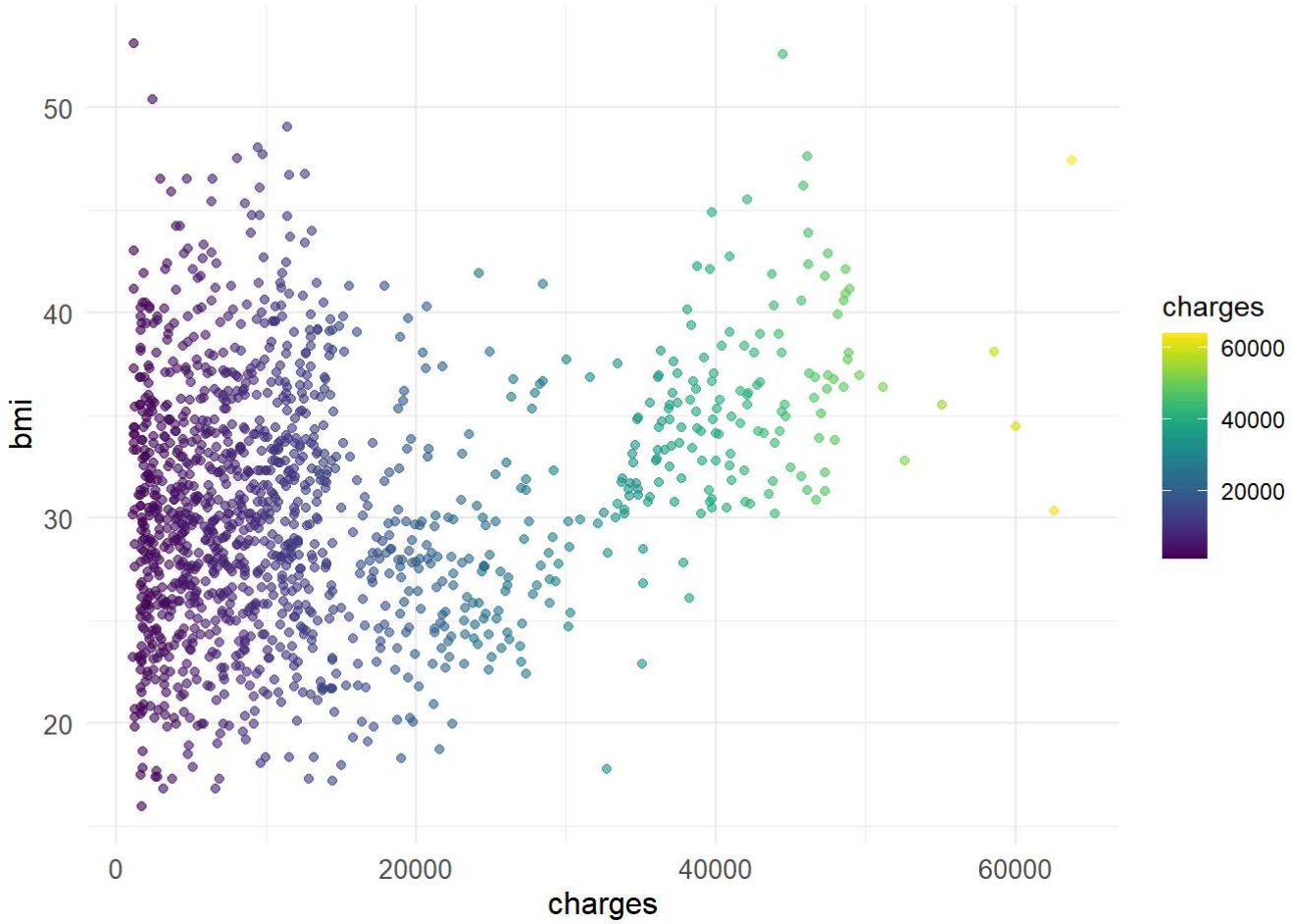

```
library(ggthemes)
library(viridis)
```

```
## Loading required package: viridisLite
```

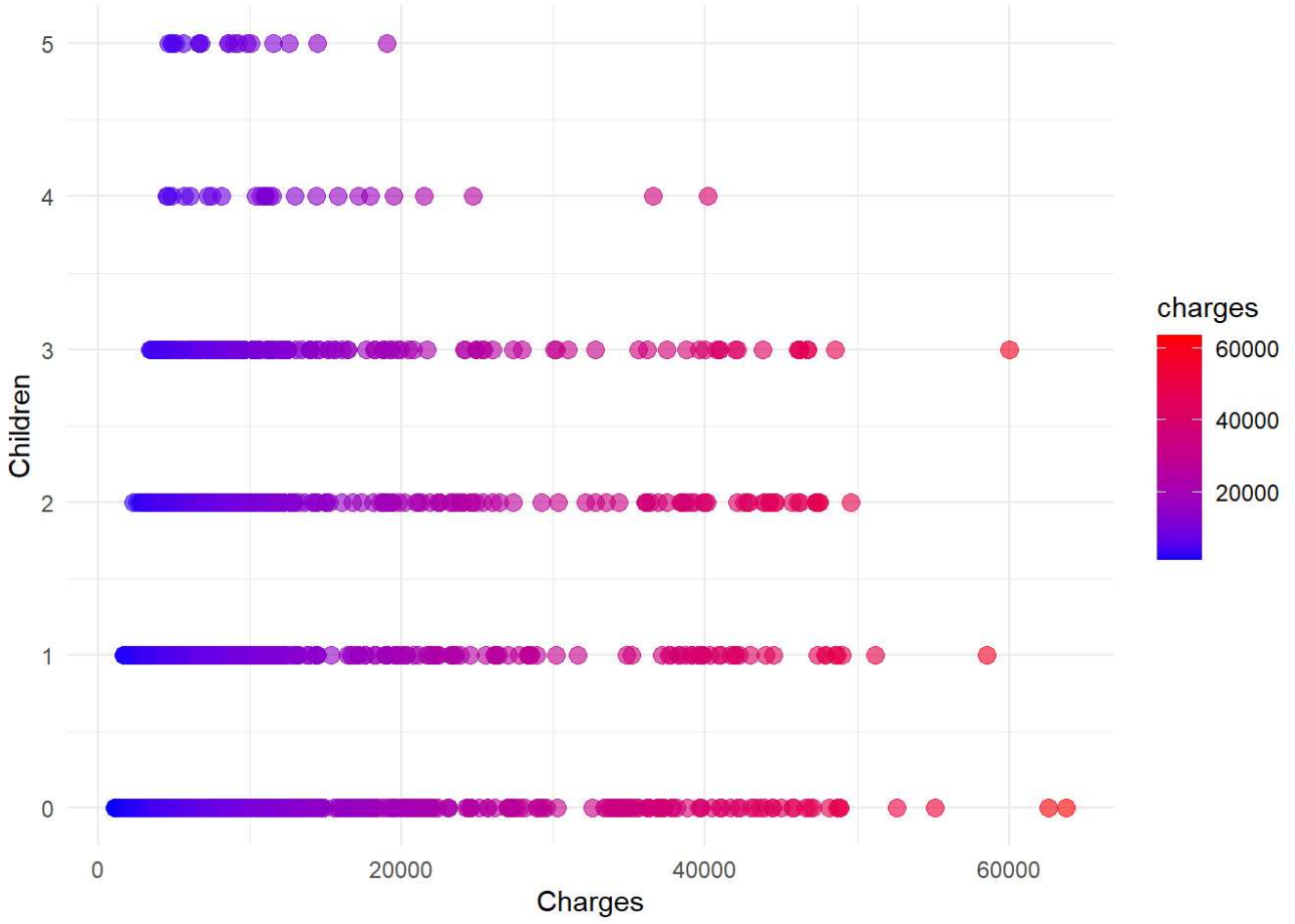
```
##
## Attaching package: 'viridis'
```

```
## The following object is masked from 'package:scales':
##
##     viridis_pal
```

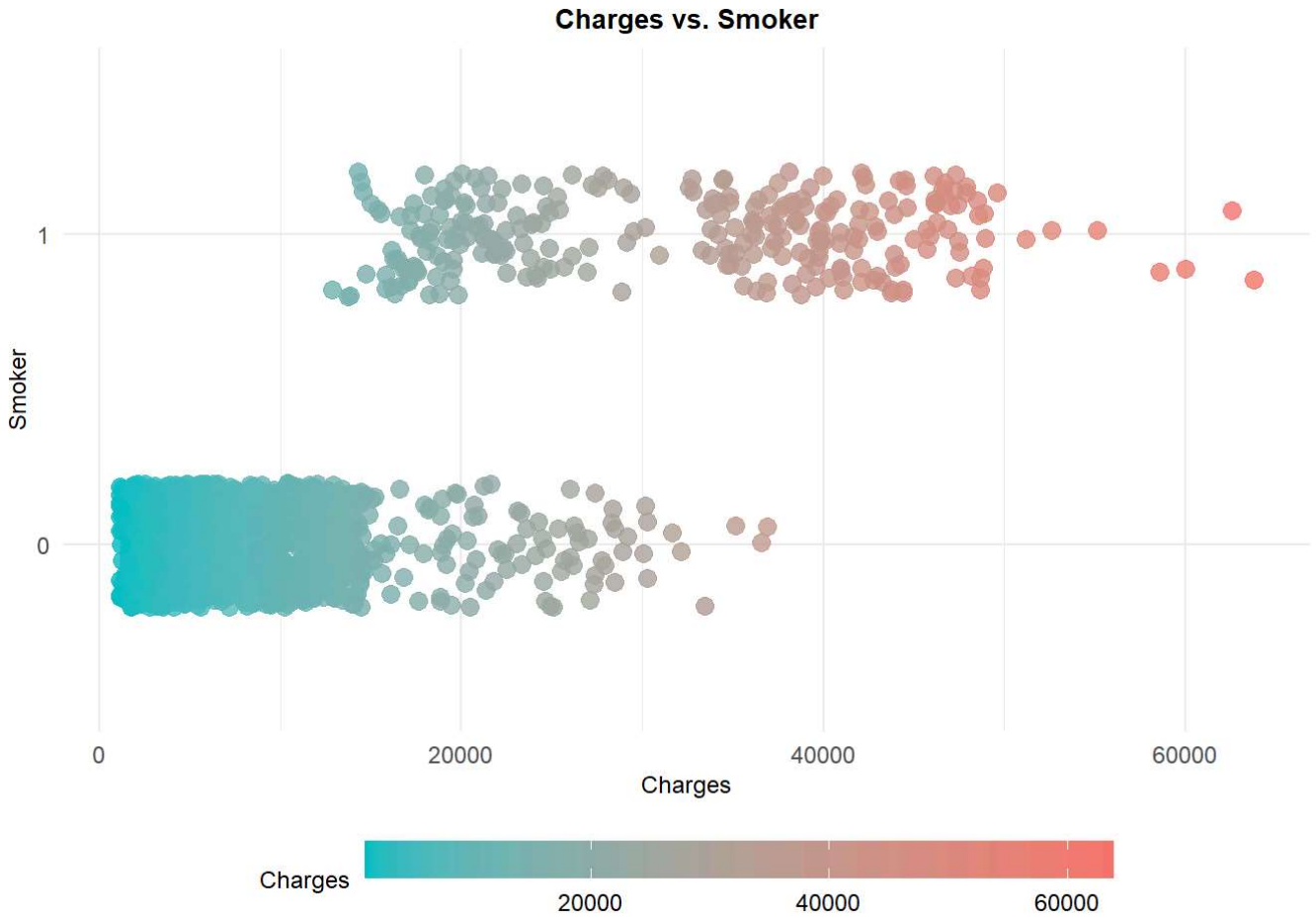
```
charges_and_bmi_scatter <- medcostnewdf %>%
  ggplot(aes(x = charges, y = bmi)) +
  geom_point(aes(color = charges), alpha = .6) +
  labs(x = "charges", y = "bmi", color = "charges") +
  scale_color_viridis() +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "right",
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

charges_and_bmi_scatter
```



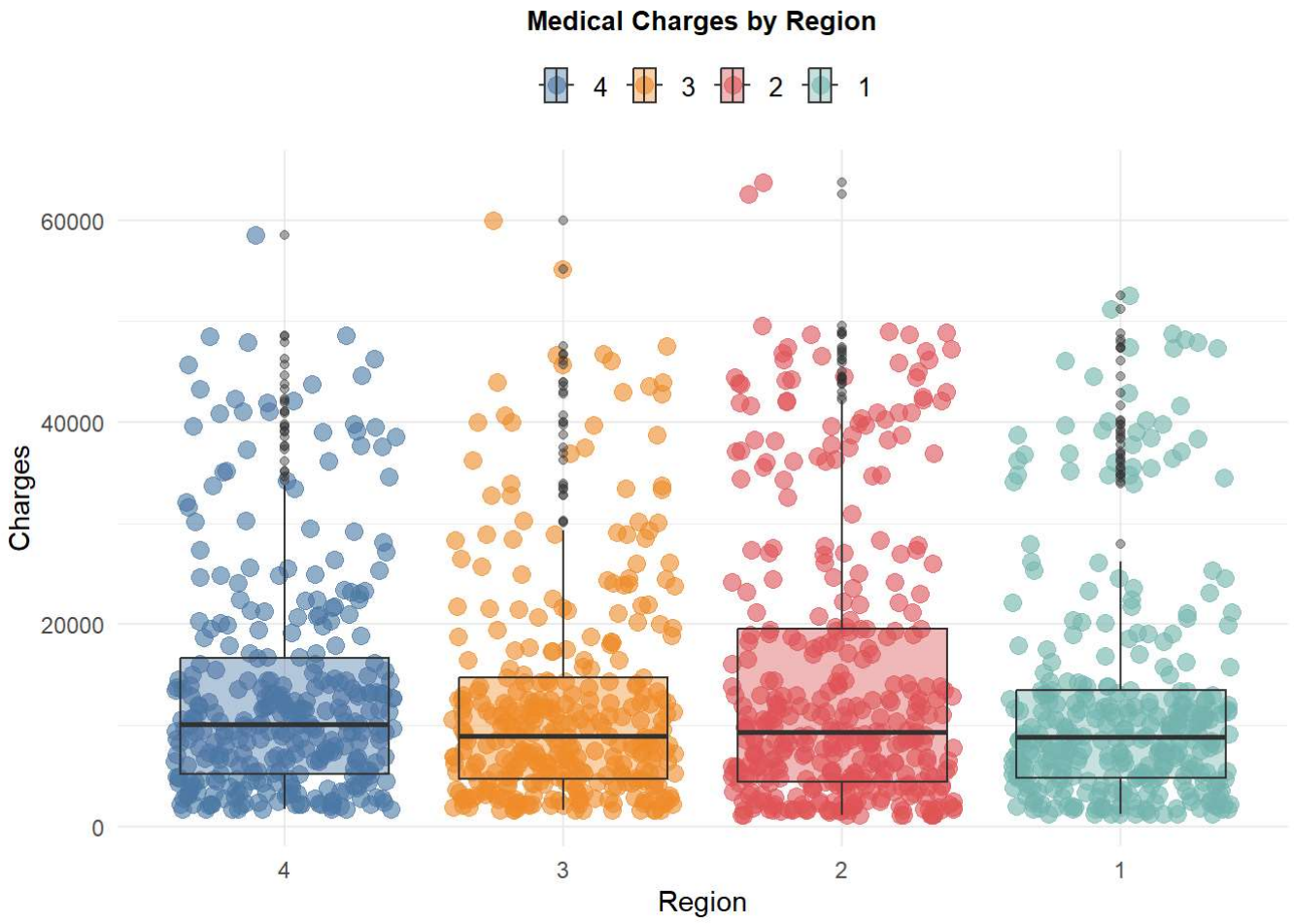

```
childrengraph<-medcostnewdf %>%
  ggplot(aes(x = charges, y = children, color = charges)) +
  geom_point(alpha = 0.6, size = 3) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(x = "Charges", y = "Children", color = "charges") +
  theme_minimal()
childrengraph
```




```
library(ggplot2)

charges_and_smoker <- ggplot(medcostnewdf, aes(x = charges, y = factor(smoker), color = charge
s)) +
  geom_jitter(width = 0.2, height = 0.2, alpha = 0.8, size = 3) +
  scale_color_gradient(low = "#00BFC4", high = "#F8766D") +
  labs(x = "Charges",
       y = "Smoker",
       title = "Charges vs. Smoker",
       color = "Charges") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5, size = 9),
        axis.title.x = element_text(size = 9),
        axis.title.y = element_text(size = 9),
        axis.text = element_text(size = 9),
        legend.position = "bottom",
        legend.title = element_text(size = 9),
        legend.text = element_text(size = 9),
        legend.direction = "horizontal",
        legend.key.width = unit(2, "cm"),
        legend.key.height = unit(0.5, "cm"))

charges_and_smoker
```

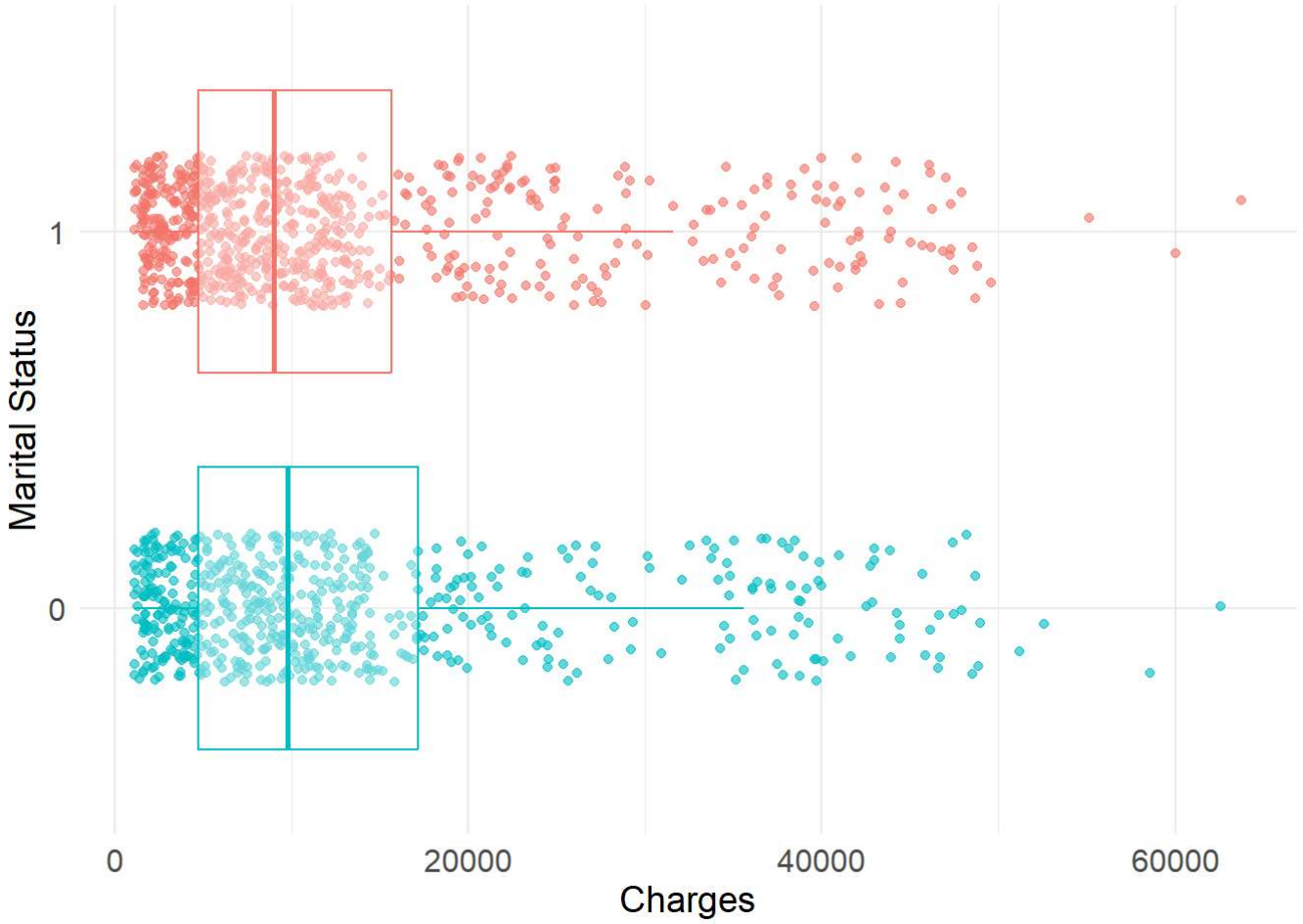



```
library(ggplot2)
library(ggthemes)

charges_and_region <- medcostnewdf %>%
ggplot(aes(x = charges, y = region)) +
  geom_jitter(aes(color = region), alpha = .6, size = 3) +
  geom_boxplot(aes(fill = region), alpha = .4) +
  scale_color_tableau() +
  scale_fill_tableau() +
  labs(x = "Charges",
       y = "Region") +
  coord_flip() +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "top",
        legend.direction = "horizontal",
        legend.justification = "center",
        legend.title = element_blank(),
        legend.text = element_text(size = 10))

charges_and_region +
  ggtitle("Medical Charges by Region") +
  theme(plot.title = element_text(size = 10, face = "bold"))
```




```
medcostnewdf$`Marital stat`  <- as.factor(medcostnewdf$`Marital stat`)
medcostnewdf <- medcostnewdf%>%
                  mutate(`Marital stat` = fct_recode(`Marital stat`,
                                      "0" = "0",
                                      "1"= "1"))
```

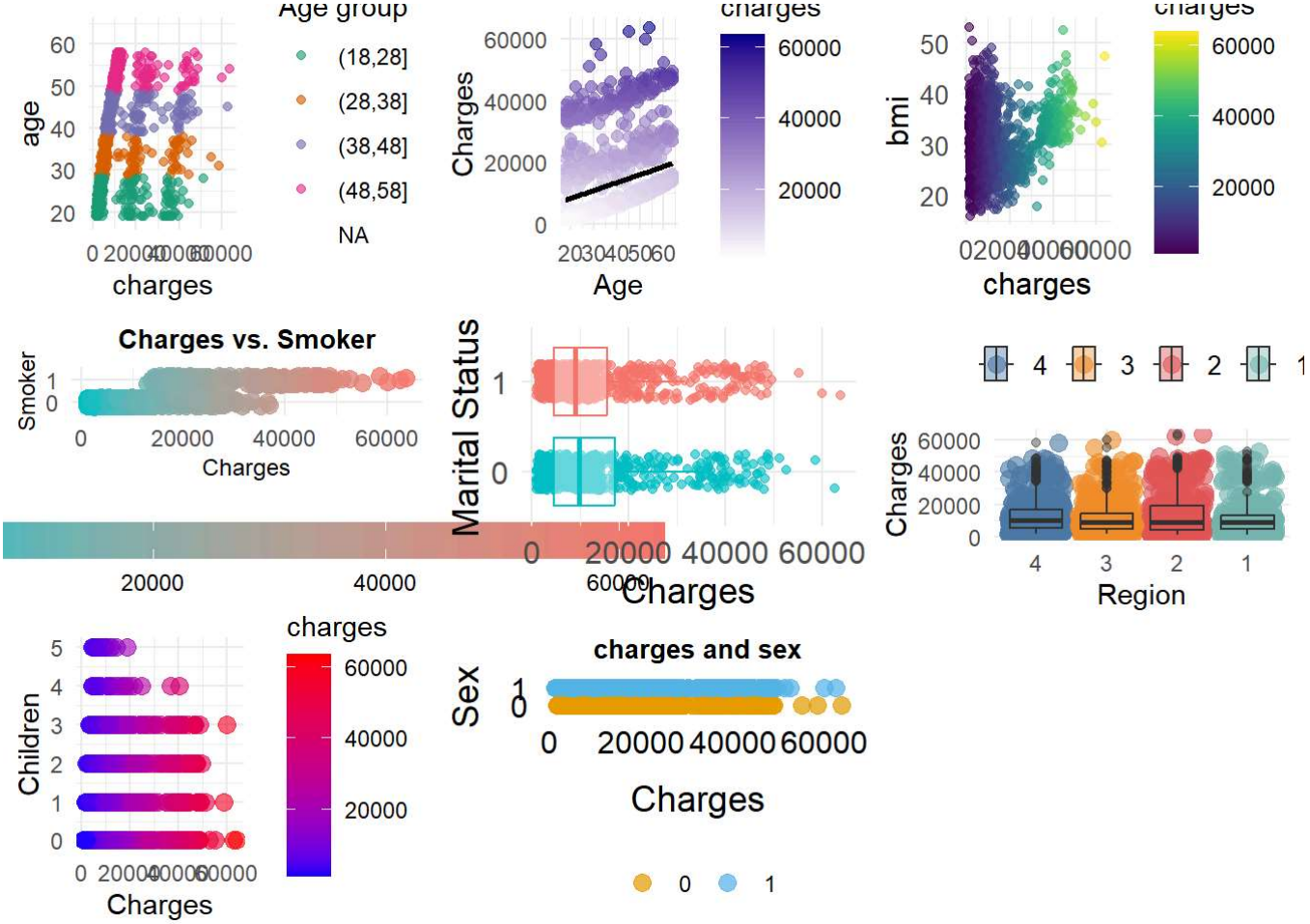
```
library(ggplot2)

charges_and_maritalstat <-ggplot(medcostnewdf, aes(x = charges, y = `Marital stat`, color = `Mar
ital stat`)) +
  geom_jitter(alpha = 0.6, height = 0.2, width = 0.1) +
  geom_boxplot(fill = "white", alpha = 0.4, outlier.shape = NA) +
  labs(x = "Charges", y = "Marital Status") +
  theme_minimal() +
  theme(legend.position = "none",
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 12)) +
  scale_color_manual(values = c("#00BFC4", "#F8766D", "#A3A500"))
charges_and_maritalstat
```




```
gridExtra::grid.arrange(agecharges,charges_and_age,charges_and_bmi_scatter,charges_and_smoker, c
harges_and_maritalstat,charges_and_region,childrengraph,charges_and_sex)
```

```
## Warning: Removed 208 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

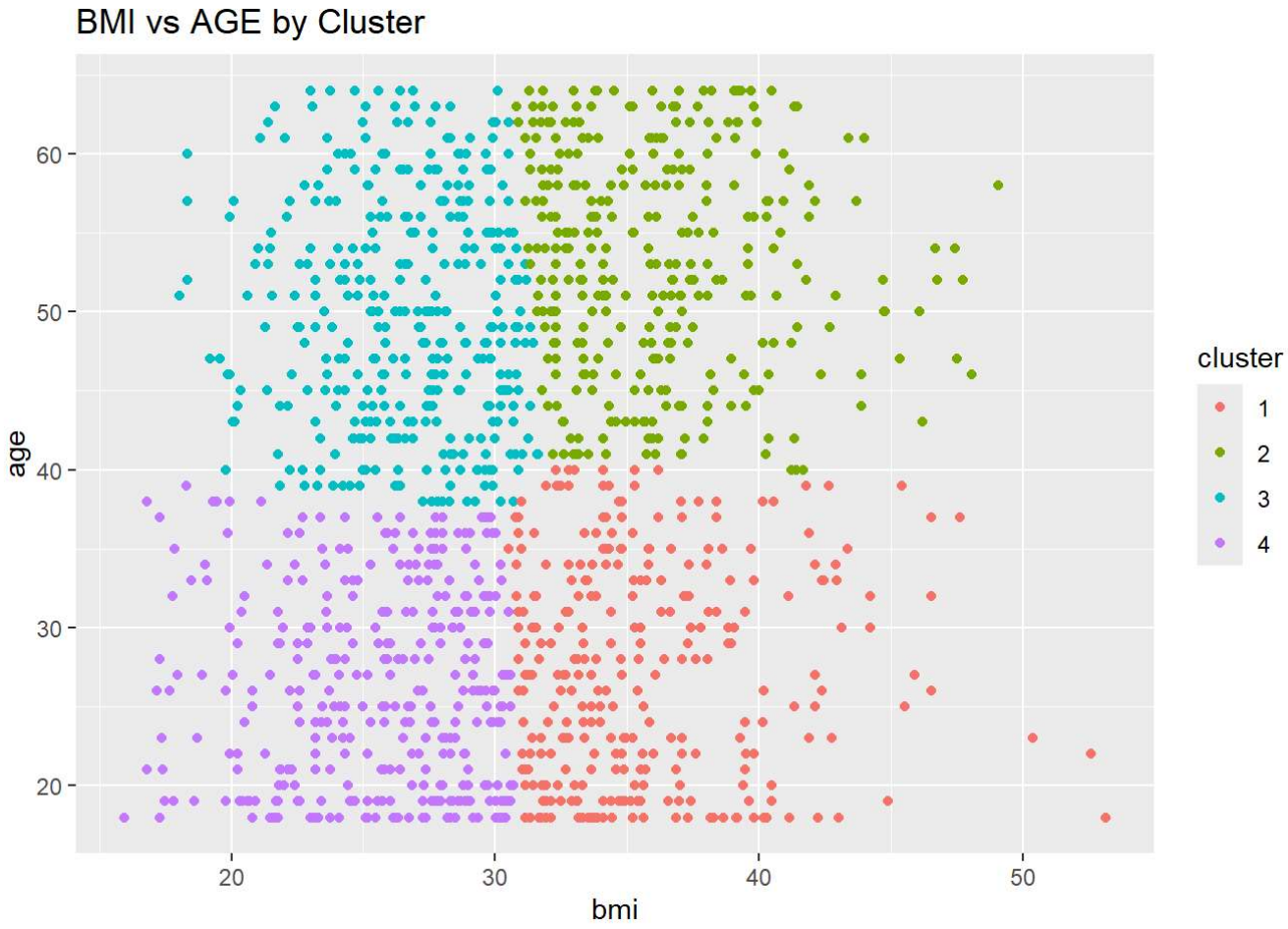


CLUSTER ANALYSIS

```
# Load the required packages
library(dplyr) # for data manipulation
library(cluster) # for cluster analysis
library(ggplot2) # for data visualization


# Subset the relevant variables
numeric_vars <- c("bmi", "age")
exclude_vars <- c("sex", "region", "children", "Marital.stat", "smoker")
data_subset <- medcostnewdf %>% select(-one_of(exclude_vars))
```

```
## Warning: Unknown columns: `Marital.stat`
```

```
# Normalize the numeric variables
data_scaled <- scale(data_subset[, numeric_vars])

# Perform k-means clustering with k=3
set.seed(123)
kmeans_output <- kmeans(data_scaled, centers = 4)

# Extract the cluster assignments
cluster_assignments <- kmeans_output$cluster

# Add the cluster assignments to the original dataset
data_clustered <- data_subset %>%
  mutate(cluster = as.factor(cluster_assignments))

# Visualize the clusters using scatter plot
ggplot(data_clustered, aes(x = bmi, y = age, color = cluster)) +
  geom_point() +
  labs(title = "BMI vs AGE by Cluster", x = "bmi", y = "age")
```



```
# Compute the average charges by cluster
data_summary <- data_clustered %>%
  group_by(cluster) %>%
  summarize(avg_charges = mean(charges))

# Print the summary statistics
print(data_summary)
```

```
## # A tibble: 4 × 2
##   cluster avg_charges
##   <fct>         <dbl>
## 1 1            12949.
## 2 2            18419.
## 3 3            14197.
## 4 4             7989.
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
medcostnewdf<-na.omit(medcostnewdf)

set.seed(500) ## chosen arbitrarily; helps with replication across runs
inTraining <- createDataPartition(medcostnewdf$charges , ## indicate the outcome - helps in bala
ncing the partitions
                                  p = .90, ## proportion used in training+ testing subset
                                  list = FALSE)
training <-medcostnewdf[ inTraining,]
holdout  <-medcostnewdf[-inTraining,]

## centering and scaling as part of the pre-processing step
preProcValues <- preProcess(training, method = c("center", "scale"))
#preProcValues <- preProcess(training, method = "pca")
preProcValues
```

```
## Created from 1206 samples and 8 variables
##
## Pre-processing:
##   - centered (4)
##   - ignored (4)
##   - scaled (4)
```

```
## Next, create the scaled+centered of the training+testing subset of the dataset
trainTransformed <- predict(preProcValues,training)
trainTransformed
```

```
## # A tibble: 1,206 × 8
##       age sex      bmi children smoker region charges `Marital stat`
##     <dbl> <fct>  <dbl>    <dbl> <fct>  <fct>    <dbl> <fct>
##  1 -1.44  0     -0.459  -0.920  1      1        0.295 0
##  2 -1.51  1      0.497  -0.0920 0      2       -0.953 1
##  3 -0.796 1      0.372   1.56   0      2       -0.729 1
##  4 -0.441 1     -1.31   -0.920  0      3        0.715 1
##  5 -0.512 1     -0.299  -0.920  0      3       -0.777 1
##  6 -0.583 0     -0.811  -0.920  0      2       -0.786 0
##  7  0.483 0      0.444  -0.0920 0      2       -0.417 0
##  8 -0.156 0     -0.485   1.56   0      3       -0.496 1
##  9 -0.156 1     -0.144   0.736  0      4       -0.568 0
## 10  1.48  0     -0.794  -0.920  0      3        1.29  1
## # ℹ 1,196 more rows
```

```
## apply the same scaling and centering on the holdout set, too
holdoutTransformed <- predict(preProcValues,holdout)
holdoutTransformed
```

```
## # A tibble: 132 × 8
##       age sex      bmi children smoker region charges `Marital stat`
##     <dbl> <fct>  <dbl>    <dbl> <fct>  <fct>    <dbl> <fct>
##  1 -1.15  1      0.600  -0.920  0      1      -0.945  1
##  2 -1.44  1     -0.996  -0.0920 0      1      -0.944  0
##  3  1.12  0      0.335   0.736  0      3      -0.0851 0
##  4 -1.51  0     -0.717  -0.920  0      4      -0.914  1
##  5 -1.08  0     -0.671  -0.920  0      4      -0.844  1
##  6  1.12  1      1.07   -0.920  0      1       0.603  0
##  7 -0.228 1      0.730  -0.0920 1      2       2.09   0
##  8 -1.29  0      0.475   0.736  0      3      -0.800  1
##  9 -0.228 1      0.605  -0.920  1      2       2.01   0
## 10  1.34  0      0.180   0.736  0      4       0.0251 0
## # ℹ 122 more rows
```

```
fitControl <- trainControl(method = "repeatedcv", ## indicate that we want to do k-fold CV
                           number = 10, ## k = 10
                           repeats = 10) ## and repeat 10-fold CV 10 times
```

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10)
set.seed(500)
knn_fit <- train(charges ~ ., data = trainTransformed, method = "knn",
                 metric= 'RMSE',
                 trControl=trctrl,
                 preProcess = c("center", "scale"),
                 tuneLength = 10)

knn_fit
```

```
## k-Nearest Neighbors
##
## 1206 samples
##    7 predictor
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1085, 1085, 1084, 1084, 1086, 1086, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    5  0.4804083  0.7678491  0.3001548
##    7  0.4794734  0.7694764  0.3053422
##    9  0.4817726  0.7682249  0.3069432
##   11  0.4841812  0.7668182  0.3100688
##   13  0.4900269  0.7627152  0.3162788
##   15  0.4948392  0.7597924  0.3186572
##   17  0.5007970  0.7563860  0.3220995
##   19  0.5060940  0.7541346  0.3251238
##   21  0.5122426  0.7526934  0.3292049
##   23  0.5176215  0.7525875  0.3320558
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 7.
```

```
## make predictions on the hold-out set
predvals <- predict(knn_fit, holdoutTransformed)

## compute the performance metrics
postResample(pred = predvals, obs = holdoutTransformed$charges)
```

```
##      RMSE  Rsquared       MAE
## 0.4536478 0.7828141 0.2802838
```

```
#varImp(knn_fit)
```

###############LASSO FIT###

```
set.seed(500)
fitControl <- trainControl(method = "repeatedcv", ## indicate that we want to do k-fold CV
                           number = 10, ## k = 10
                           repeats = 10) ## and repeat 10-fold CV 10 times

lassofit<- train(charges ~ .,
                    data = trainTransformed,
                    method = "glmnet",
                  metric = 'RMSE',
                trControl=fitControl,
                preProc = c("center","scale"),
                tuneGrid = expand.grid(alpha = 1,
                                       lambda = 0))
lassofit
```

```
## glmnet
##
## 1206 samples
##    7 predictor
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1085, 1085, 1084, 1084, 1086, 1086, ...
## Resampling results:
##
##   RMSE       Rsquared   MAE
##   0.5074758  0.7417771  0.3505153
##
## Tuning parameter 'alpha' was held constant at a value of 1
## Tuning
##  parameter 'lambda' was held constant at a value of 0
```

```
## make predictions on the hold-out set
predvals <- predict(lassofit, holdoutTransformed)

## compute the performance metrics
postResample(pred = predvals, obs = holdoutTransformed$charges)
```

```
##      RMSE  Rsquared       MAE
## 0.4309386 0.8029949 0.3044066
```

```
varImp(lassofit)
```

```
## glmnet variable importance
##
##                  Overall
## smoker1         100.0000
## age              38.0142
## bmi              20.3790
## children          5.9919
## region1           3.4727
## region2           3.1732
## `Marital stat`1   1.7024
## region3           0.5842
## sex1              0.0000
```

```
#multiple LINEAR REGRESSION
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10)
set.seed(500)
mlr_fit <- train(charges ~ ., data = trainTransformed,
                 method = "lm",
                 trControl=trctrl,
                 metric= "RMSE",
                 preProcess = c("center", "scale"),
                 tuneLength = 10)

mlr_fit
```

```
## Linear Regression
##
## 1206 samples
##    7 predictor
##
## Pre-processing: centered (9), scaled (9)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1085, 1085, 1084, 1084, 1086, 1086, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   0.507559  0.7416874  0.3507414
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
## make predictions on the hold-out set
predvals <- predict(mlr_fit, holdoutTransformed)

## compute the performance metrics
postResample(pred = predvals, obs = holdoutTransformed$charges)
```

```
##      RMSE  Rsquared       MAE
## 0.4309676 0.8029312 0.3045746
```

```
varImp(mlr_fit)
```

```
## lm variable importance
##
##                        Overall
## smoker1               100.0000
## age                    37.8899
## bmi                    19.4753
## children                5.9620
## region1                 3.0457
## region2                 2.7249
## `\\`Marital stat\\`1`   1.6642
## region3                 0.6846
## sex1                    0.0000
```

##############################randomforest###

```
library(caret)
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
fitControl <- trainControl(method = "repeatedcv", ## indicate that we want to do k-fold CV
                           number = 10, ## k = 10
                           repeats = 10) ## and repeat 10-fold CV 10 times


rf_random <- train(charges ~ .,
                   data = trainTransformed, ## the dataset containing the training-testing subset
                   method = "rf",
                   metric = 'RMSE',
                   trControl = fitControl)
rf_random
```

```
## Random Forest
##
## 1206 samples
##    7 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 1085, 1086, 1086, 1085, 1085, 1086, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE       Rsquared   MAE
##   2     0.4489494  0.8339313  0.3144038
##   5     0.3890889  0.8440815  0.2258985
##   9     0.3998851  0.8360636  0.2321422
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 5.
```

```
## make predictions on the hold-out set
predvals <- predict(rf_random, holdoutTransformed)

## compute the performance metrics
postResample(pred = predvals, obs = holdoutTransformed$charges)
```

```
##      RMSE  Rsquared       MAE
## 0.3064588 0.9042407 0.1825966
```

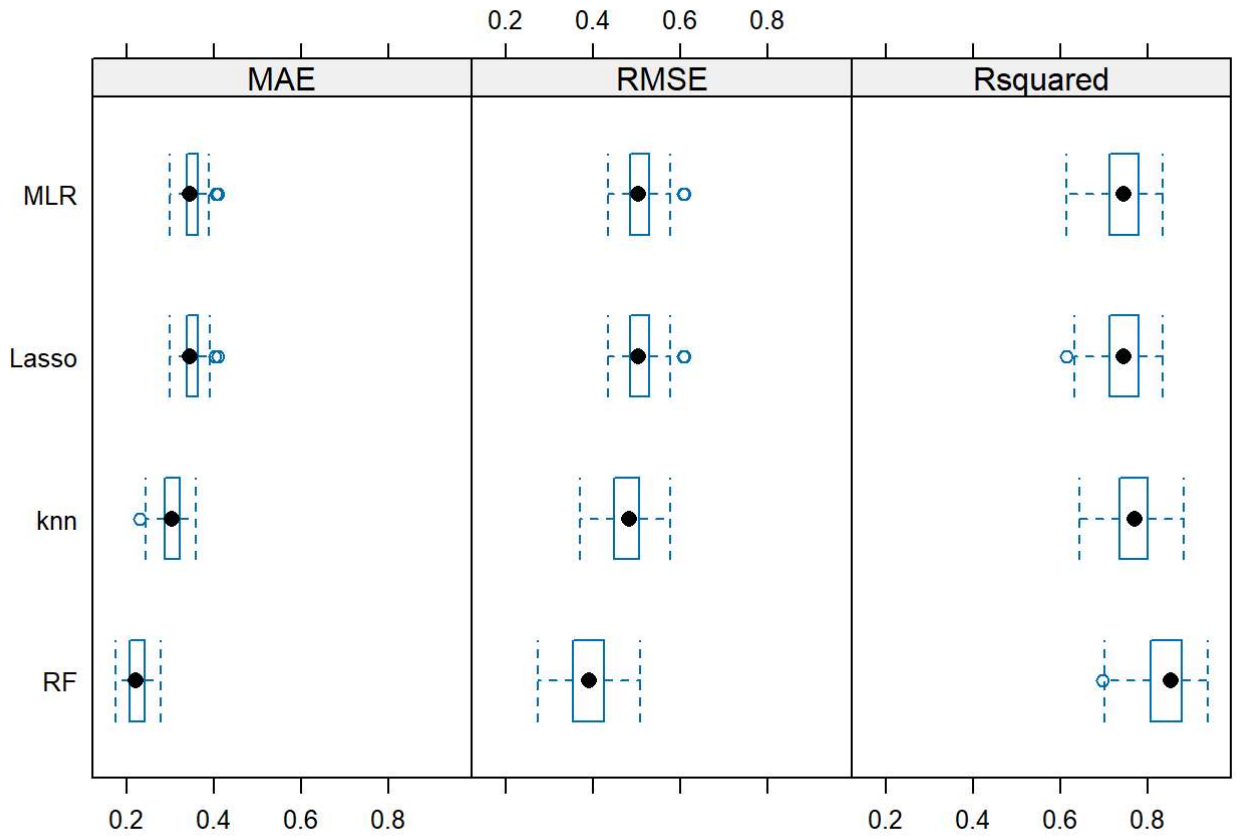
```
varImp(rf_random)
```

```
## rf variable importance
##
##                   Overall
## smoker1         100.00000
## bmi              27.01802
## age              21.19666
## children          2.27451
## region2           0.26496
## sex1              0.16270
## region3           0.12303
## `Marital stat`1   0.04515
## region1           0.00000
```

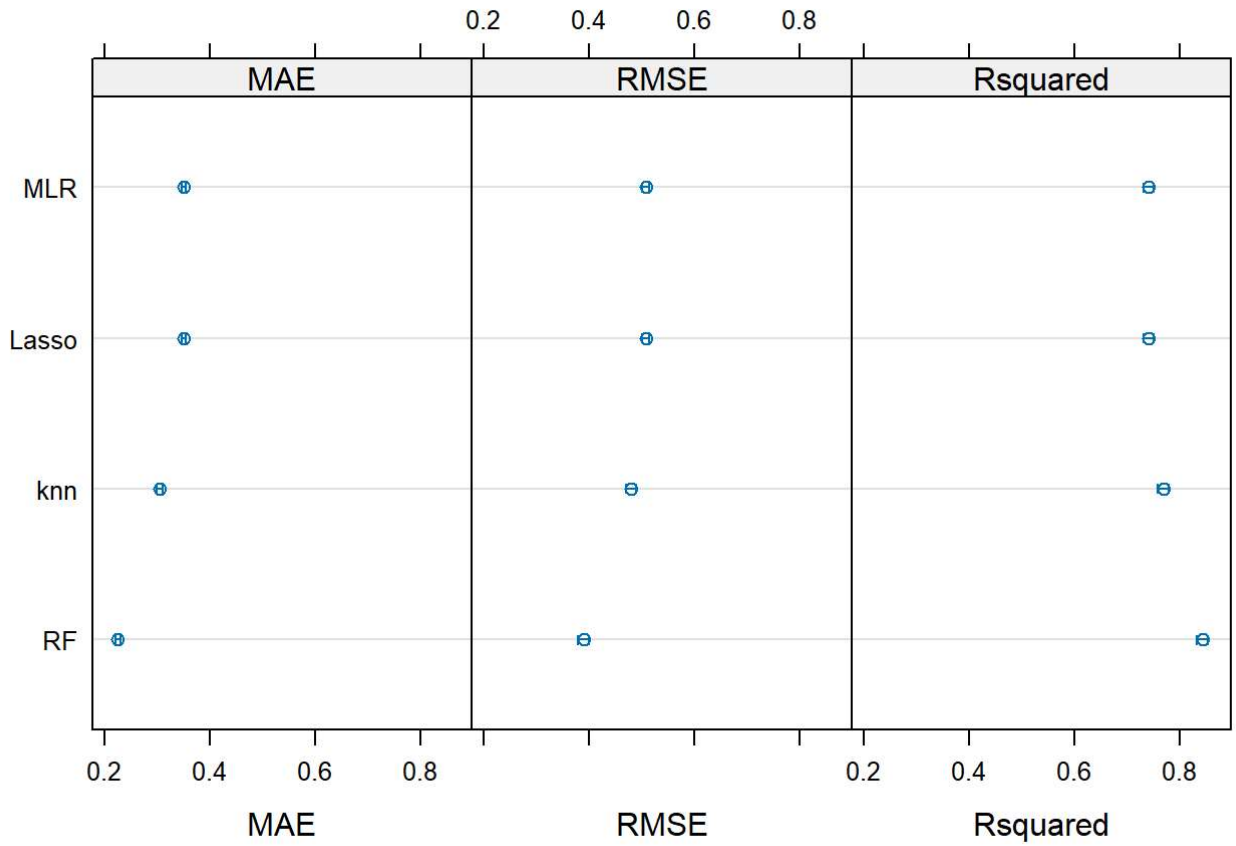
```
library(mlbench)
library(caret)
results <- resamples(list(RF=rf_random, MLR=mlr_fit, Lasso=lassofit, knn=knn_fit))
# summarize the distributions
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: RF, MLR, Lasso, knn
## Number of resamples: 100
##
## MAE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## RF    0.1754798 0.2080370 0.2223493 0.2258985 0.2431380 0.2791636    0
## MLR   0.3000417 0.3379081 0.3461980 0.3507414 0.3637401 0.4099048    0
## Lasso 0.2996729 0.3381513 0.3461811 0.3505153 0.3639544 0.4097753    0
## knn   0.2314996 0.2885404 0.3041937 0.3053422 0.3229828 0.3598666    0
##
## RMSE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## RF    0.2721212 0.3547925 0.3899213 0.3890889 0.4238484 0.5079948    0
## MLR   0.4346483 0.4840083 0.5027159 0.5075590 0.5272156 0.6089384    0
## Lasso 0.4350093 0.4844766 0.5024495 0.5074758 0.5269821 0.6081381    0
## knn   0.3693993 0.4491282 0.4823476 0.4794734 0.5055697 0.5758737    0
##
## Rsquared
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## RF    0.6970479 0.8090122 0.8532380 0.8440815 0.8798160 0.9391421    0
## MLR   0.6140089 0.7145260 0.7455699 0.7416874 0.7811615 0.8347825    0
## Lasso 0.6140302 0.7146389 0.7450505 0.7417771 0.7805457 0.8346329    0
## knn   0.6450131 0.7374851 0.7702387 0.7694764 0.8017094 0.8832337    0
```

```
# boxplots of results
bwplot(results)
```




```
# dot plots of results
dotplot(results)
```



Confidence Level: 0.95