Predicting High Healthcare Costs

Myron Keith Gibert Jr

2022-11-07

Contents

Introduction **1**
- Research Question . . . 1
- Setup . . . 1
- Read data . . . 1
- Modify factor classes . . . 2

Exploring whether any variable strongly correlates with high patient expenses. **2**
- Write a function to generation correlation matrix for plotting . . . 2
- Simple correlation plot . . . 3
- Multiple correlation plot (Charges vs Smoker+Other Variables) . . . 5

Modeling Healthcare costs using Age, BMI, and Smoker status and comparing to Simple Regression using only Smoker status. **6**
- Split data into testing and training sets . . . 6
- Construct a multiple regression model using Age, BMI, and Smoker status to predict cost . . . 7

Test model **7**
- Write Equation to Extract Regression Equation . . . 8

Evaluate Model **8**
- Visualize Predicted vs Actual Results on the Testing Set . . . 8
- Generate Alternative Models . . . 10
  - Alternative Model 1: Simple regression model using Smoker status to predict cost . . . 10
  - Alternative Model 2: Multiple regression model using BMI + Smoker status to predict cost . . . 11
- Compare Testing Model for Healthcare Costs to alternatives using R-Squared Method . . . 11
  - Combine model results . . . 12
  - Plot results . . . 13

Discussion **14**
- Improving the model . . . 14
  - Collect more data to increase the number of variables or observations for each variable . . . 14
  - Normalize the data . . . 14
  - Add regularization to the model . . . 14
  - Minimize the effect of small sample sizes . . . 14
- Conclusions . . . 14
- Correspondence . . . 14

Introduction

I am writing this document as a coding sample submission to demonstrate the following skills:

- Transferable comfort with organized data (rows x columns) of any kind
- Statistical Data Analysis of categorical and continuous variables
- Competency in R
- Reproducible research using RMarkdown
- Use of statistical regression (Simple and multiple)
- Data Visualization
- Writing Functions in R
- Looping functions

To accomplish this, I will be using Health Insurance downloaded from Kaggle a repository of publicly available data sets for machine learning and statistical modelling.

These data include patient information across seven variables:

- Age
- Sex
- Body Mass Index (BMI)
- Children (# of)
- Smoker (Yes they smoke or no they do not)
- Region (of Service)
- Charges (Healthcare costs)

The repository and relevant materials for this project can be downloaded from here

While I'm using health insurance data for this example, I am able to perform a similar analysis on any data set to answer similar questions in any field.

For example, with access to the right data, I could also answer these questions using a similar methodology:

- What genes are likely to predict patient outcome? (Biology)
- What policy actions lead to less prison recidivism? (Criminal Justice)
- What is the best predictor of financial success or workplace productivity? (Business)
- What beliefs are best able to predict a person's political ideology? (Political Science)
- What policy actions lead to better student performance? (Education)
- How likely is it to rain tomorrow compared to the previous day? (Meteorology)

Research Question

What variable or combination of variables, if any, can accurately predict Healthcare costs (charges)?

Setup

I will be using the tidyverse and reshape packages for general data wrangling and manipulation. The ggplot2 and ggthemes packages are used for visualization. Lastly, rlang is a dependency for tidymodels, the package that I'll be using for statistical modeling of the data.

This setup chunk will set chunk options. Then, it installs packages (if needed) and then loads them into R.

Read data

Now, I'll read the data. The data is in .csv format, so we'll load in using read.csv. Then, we'll look at the first five rows of the data.

```
# Read in the csv file.  Set the first line of the file as the header for the table
```

```
data <- read.csv("insurance.csv",header=TRUE)

head(data,5)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
```

I want age, bmi, children, and charges to be numeric, so let's convert them.

Modify factor classes

I used dplyr's mutate function and piping (%>%) to coerce the required columns to numeric values. Then I used sapply to extract the object class for each column, showing that all columns are each numeric or character.

```
# Coerce columns to numeric using dplyr mutate and as.numeric()
data_fixed <- data %>%
        mutate(age = as.numeric(age),
               bmi = as.numeric(bmi),
               children = as.numeric(children),
               charges = as.numeric(charges))

# Extract object classes using sapply
sapply(data_fixed,class)
```

```
##         age         sex         bmi    children      smoker      region
##   "numeric" "character"   "numeric"   "numeric" "character" "character"
##     charges
##   "numeric"
```

Looks good. Let's keep going.

Exploring whether any variable strongly correlates with high patient expenses.

As per my research question, I will investigate whether any variable correlates with high patient costs. I will use the pearson correlation for linearity, as I believe that this will serve me better for downstream modeling.

Write a function to generation correlation matrix for plotting

To accomplish this, I will utilize a function that will process my data set. It will first reorganize that correlation matrix by hierarchical clustering. This will cluster similar groups together and help magnify any trends in the data. Then it will delete the upper triangle of the matrix to remove duplicate values and increase readability.

```
# Initialize the function
process_cormat <- function(x){

# Generate distances
dd <- as.dist((1-x)/2)
```

```
# Generate hierarchical clustering object
hc <- hclust(dd)

# Reorganize matrix by clustering object
x <-x[hc$order, hc$order]

# Replace upper half of matrix with NAs
x[upper.tri(x)] <- NA

# Return final result
  return(x)

}
```

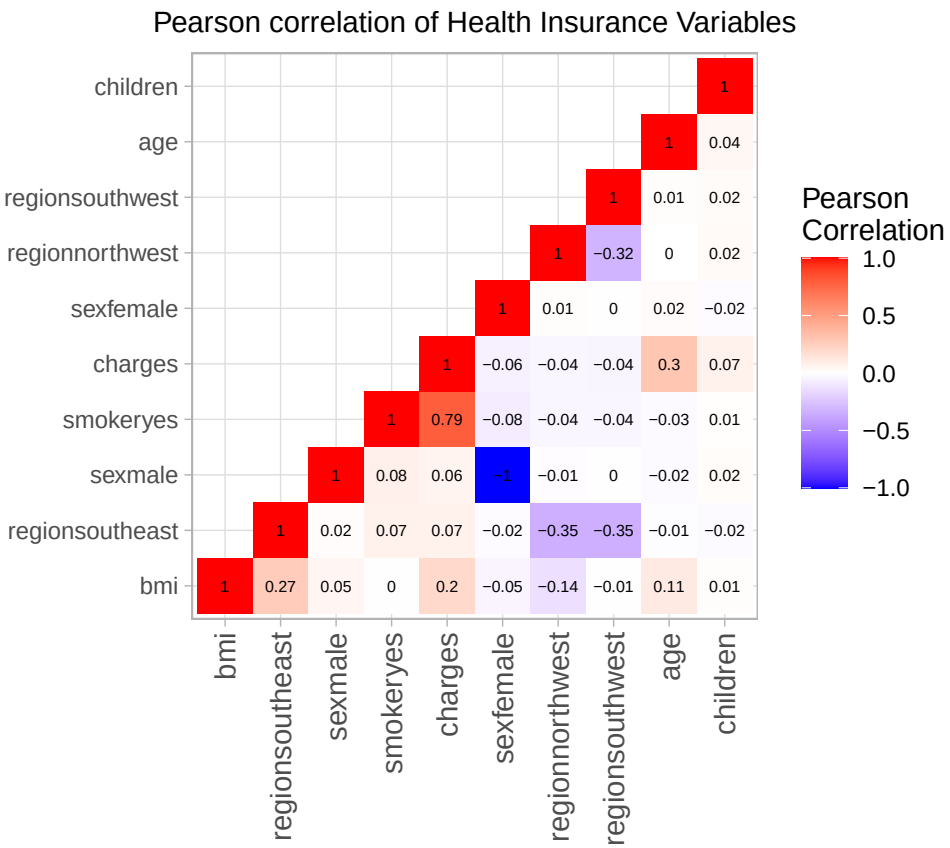
Simple correlation plot

Now, I will generate a simple correlation plot, format it using the function that I created in the previous section, melt using reshape2 to convert from wide format to tall and then visualize the data using the ggplot2 package and geom_tile().

Red values are positively correlated, meaning as one variable increases, the other increases as well. The higher the correlation, the straighter the line and stronger the relationship. Negative correlations occur when one variable increases and the other decreases. Strong correlations are closer to 1 or -1, while weak correlations are closer to 0.3 and -0.3.

```
# Generate correlation matrix using ~variable and 0+ to add intercept.

corrdata <- model.matrix(~0+., data=data_fixed) %>%
  cor(use="pairwise.complete.obs")

# Process correlation matrix using formula from previous section.

corrdata_processed <- process_cormat(corrdata)

# Use melt to convert the data from "wide" format to "tall" format.

corrdata_tall <- melt(corrdata_processed, na.rm = TRUE)

## PLOT FUNCTION

# Initialize ggplot2, with the variables on the x and y axes and the
# value of the fill visualizing the correlation

p <- ggplot(corrdata_tall,aes(Var1,Var2,fill=as.numeric(value))) +

# Use geom_tile to visualize heatmap
  geom_tile() +

# Add data labels using geom_text and round them to 2 digits
  geom_text(size=rel(2.0),aes(label=round(as.numeric(value),2))) +

# Add plot title
  ggtitle("Pearson correlation of Health Insurance Variables") +
```

```
# Use the light theme
  theme_light() +

# High correlations are red, low are blue, and mid are white.
# Then, provide a name for the legend that will appear.
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +

# Modify the formatting of the figure text using the theme
  theme(axis.text.x=element_text(size=rel(1.2),
                                 angle = 90,
                                 vjust = 0.5,
                                 hjust = 1),
        plot.title = element_text(size=rel(1.0),
                                  hjust = 0.5),
        axis.title=element_blank()) +

# Fix the coordinates of all values for readability
  coord_fixed()

## END PLOT FUNCTION
# Return the plot
p
```



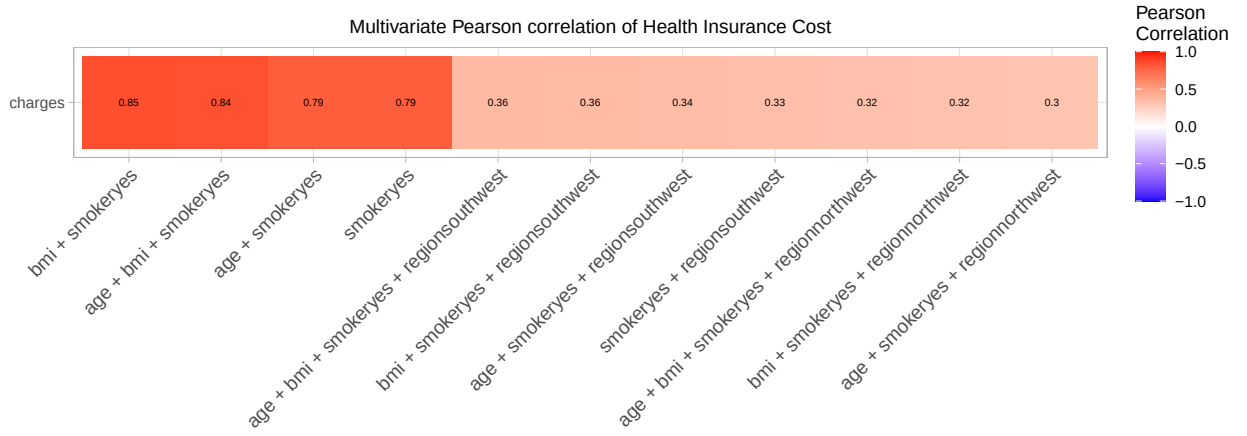
Being a smoker is fairly strongly correlated with higher health costs (Very red, cor = 0.79). Age and BMI

have weaker correlations (~0.3). These make intuitive sense at a glance.

Now, let's take a look at some combinations!

Multiple correlation plot (Charges vs Smoker+Other Variables)

```
## Using the model matrix function. ~ to set comparison, then list variables,
## 0+ to include intercept (omit if not needed), and "." to indicate that we
## wish to compare all variable combinations against the listed ones.

corrdata <- model.matrix(~0+charges*age*bmi*smoker*., data=data_fixed) %>%
  cor(use="pairwise.complete.obs")

# Process this matrix using our function

corrdata_processed <- process_cormat(corrdata)

# Convert to tall format

corrdata_tall <- melt(corrdata_processed, na.rm = TRUE) %>%

# Increase readibility of variable combinations and remove combinations that
# include the charge variable

  mutate(detector = str_detect(Var1,"charges"),Var3 = str_replace_all(Var1,":"," + ")) %>%

# Only keep variables that are actually correlated.
  filter(Var2 == "charges" & detector == FALSE & abs(value) >= 0.3)

# Visualize plot using ggplot2 like before
p <- ggplot(corrdata_tall,aes(reorder(Var3,desc(value)),Var2,fill=as.numeric(value))) +
  geom_tile() +
  geom_text(size=rel(2.0),aes(label=round(as.numeric(value),2))) +
  ggtitle("Multivariate Pearson correlation of Health Insurance Cost") +
  theme_light() +
  scale_color_colorblind() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme(axis.text.x=element_text(size=rel(1.2),
                                 angle = 45,
                                 hjust = 1),
        plot.title = element_text(size=rel(1.0),
                                  hjust = 0.5),
        axis.title=element_blank()) +
  coord_fixed()

p
```



Having a high BMI while being a older smoker seems to correlate very highly with having high health costs. However, these variables might not be worth including over just looking at smoking alone, since it only leads to a slight increase in the correlation. Also, having a high BMI while being a smoker is most correlated with high healthcare costs, so maybe those two variables alone are the best predictors.

Modeling Healthcare costs using Age, BMI, and Smoker status and comparing to Simple Regression using only Smoker status.

Let's use regression modeling using the three variables (age,bmi,smoker status) and compare the model to a simple regression model for smoker status alone or BMI+Smoker to see if using all which model is actually more predictive of healthcare costs.

Split data into testing and training sets

I'm going to split the data into two subsets. One will be used to "train" the model and get it ready for prediction. The other will be used to "test" the predictive power of the model. I will first set the random seed to increase reproducibility, and then subset the data.

```
# Set random seed for reproducibility

set.seed(1234)

# Split data into training and test data sets.

data_subset_split <- initial_split(data_fixed,prop=0.80)

# Extract the training dataset

data_subset_training <- training(data_subset_split)
```

Construct a multiple regression model using Age, BMI, and Smoker status to predict cost

I can construct a multiple linear regression model that is trained on the training data set. I will then extract the model coefficients to assist in analyzing the data downstream.

```
# Construct model
lm_model <- linear_reg() %>%
            set_engine('lm') %>% # adds lm implementation of linear regression
            set_mode('regression')

# Fit model to training data
lm_fit <- lm_model %>%
          fit(charges ~ age+bmi+smoker, data = data_subset_training)

# Extract model parameters and return them
fitdata <- tidy(lm_fit)

fitdata
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  -10994.    1046.     -10.5  1.14e- 24
## 2 age             260.      13.5     19.3  3.29e- 71
## 3 bmi             302.      30.5      9.87 4.67e- 22
## 4 smokeryes     24088.     469.      51.4  9.19e-291
```

All three variables have a statistically significant (p.value < 0.05) effect on healthcare costs, and the intercept is also significant. This could be a good model.

Test model

To determine whether the model is actually useful, we will test the model on the test data set. We will extract the test data set from the split. Generate predicted charges for the training and test data sets using our model.

```
# Extract Test dataset from split

data_subset_testing <- testing(data_subset_split)

# Predict Healthcare Costs (charges) for training data set

charge_predictions_train <- predict(lm_fit,new_data = data_subset_training)

# Join the predictions to the training data for comparison to the actual values

charge_test_results_train <- data_subset_training %>%
  dplyr::select(age,bmi,smoker,charges) %>%
  bind_cols(charge_predictions_train)

# Predict Healthcare Costs (charges) for test data set

charge_predictions_test <- predict(lm_fit,new_data = data_subset_testing)

# Join the predictions to the test data for comparison to the actual values
```

```
charge_test_results_test <- data_subset_testing %>%
  dplyr::select(age,bmi,smoker,charges) %>%
  bind_cols(charge_predictions_test)
```

Write Equation to Extract Regression Equation

The coefficients (m) and intercepts (b) can be extracted from the model for visualization. The extracted equation will follow a typical y = b + mx format, where that are as many mx pairs as there are variables.

```
#Initialize equation
extract_eq <- function(lm_fit,digits){

# Extract parameters from model and provide the sign of each coefficient
        fitdata <- tidy(lm_fit) %>%
                mutate(operator = ifelse(estimate >= 0,"+","-"),
                       variable = ifelse(term == "(Intercept)","","x"))

# Initialize equation
        equation_result <- "y = "

        i <- 1

# For every coefficient, use for loop to sequentially paste its sign and value
# (rounded to the specified digits) to the initialized equation.

        for(i in 1:length(fitdata$term)){

        equation_result <- paste(
          equation_result,
          fitdata$operator[i],
          round(abs(fitdata$estimate[i]),digits),
          fitdata$variable[i],
          " ",sep="")

        }

# Return the full equation after looping

        equation_result
}
```

Evaluate Model

Now that we have applied the model to the test data set, we'll need to compare it to the training data set. A good model will have at least as much prediction ability for the test data set that it does for the training data set.

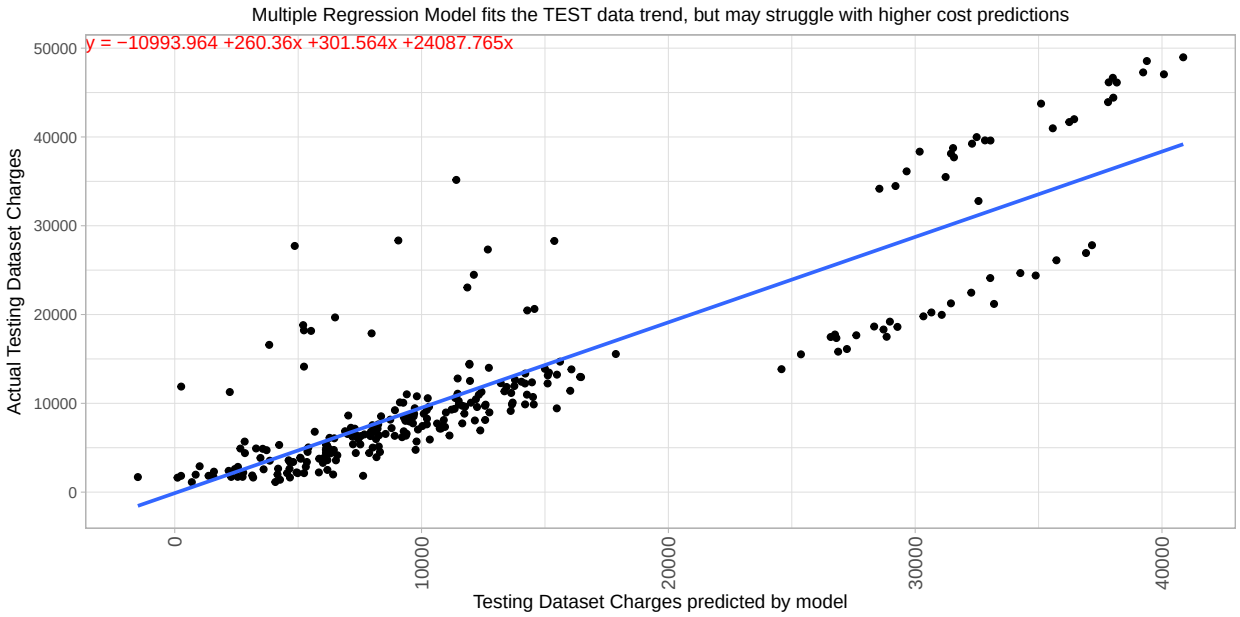
Let's see how well our model performs.

Visualize Predicted vs Actual Results on the Testing Set

First, let's see how the predicted and actual health charges compare. If the model is effective, the resulting regression line will pass through most points and few points will deviate from the line.

We'll extract the regression equation using the function from the previous section and then visualize it using ggplot2.

```
# Extract regression equation
reg_equation <- extract_eq(lm_fit,3)

# Visualize using ggplot2 comparing predicted charges (x) to actual (y)
p <- ggplot(charge_test_results_test,(aes(.pred,charges))) +

# Scatter plot
  geom_point() +

# Regression line
  geom_smooth(method = lm, se = FALSE) +

# Label axes and plot title
  xlab("Testing Dataset Charges predicted by model") +
  ylab(expression("Actual Testing Dataset Charges")) +
  ggtitle("Multiple Regression Model fits the TEST data trend,
          but may struggle with higher cost predictions") +

# Use light theme and colorblind coloring for accessibility
  theme_light() +
  scale_color_colorblind() +
  scale_fill_colorblind() +
  theme(axis.text.x=element_text(size=rel(1.2),
                                 angle = 90,
                                 vjust = 0.5,
                                 hjust = 1),
        plot.title = element_text(size=rel(1.0),
                                  hjust = 0.5),

# Remove legend
        legend.position = "none") +

# Add regression equation from first step to the top left corner of the plot
annotate('text',
         label=reg_equation,
         x=-Inf,
         y=Inf,
         hjust=0,
         vjust=1,
         color="red")

# Return the plot

p
```



It seems like the model fits very well to lower charges, but may be a little weaker on the higher end.

Generate Alternative Models

Let's compare its effectiveness to other models to determine whether Age+BMI+Smoker is the best predictor of high healthcare costs.

We'll use the R-squared value to determine the effectiveness of each model. The R-Squared value will be further explained in the "Compare Testing Model for Healthcare Costs to alternatives using R-Squared Method" section.

Alternative Model 1: Simple regression model using Smoker status to predict cost

For this model, we'll use the same methodology, but use Smoker status only to predict healthcare costs. We're not going to visualize this model for brevity, but we will extract the R-Squared value.

```
lm_model <- linear_reg() %>%
            set_engine('lm') %>% # adds lm implementation of linear regression
            set_mode('regression')

lm_fit <- lm_model %>%
          fit(charges ~ smoker, data = data_subset_training)

fitdata <- tidy(lm_fit)

fitdata
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    8514.      260.      32.8 1.12e-163
## 2 smokeryes     23871.      570.      41.9 1.78e-227
```

```
charge_predictions_smoker <- predict(lm_fit,new_data = data_subset_testing)

# Extract R-Squared value
```

```
rsq_smoker <- data_subset_testing %>%
  dplyr::select(smoker,charges) %>%
  bind_cols(charge_predictions_smoker) %>%
# Extract R-Squared value
  rsq(truth = charges, estimate = .pred) %>%
# Add identifier and keep only relevant columns
  mutate(dataset = "Test (Smoker only)") %>%
  dplyr::select(dataset,.estimate)
```

Alternative Model 2: Multiple regression model using BMI + Smoker status to predict cost

This time we'll look at BMI+Smoker, since that had the highest correlation with Healthcare costs.

```
#BMI + Smoker

lm_model <- linear_reg() %>%
            set_engine('lm') %>% # adds lm implementation of linear regression
            set_mode('regression')

lm_fit <- lm_model %>%
          fit(charges ~ bmi+smoker, data = data_subset_training)

fitdata <- tidy(lm_fit)

fitdata
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)   -2657.    1105.      -2.41 1.63e-  2
## 2 bmi             366.      35.2     10.4  4.24e- 24
## 3 smokeryes     23802.     544.      43.8  1.80e-240
```

```
charge_predictions_bmismoker <- predict(lm_fit,new_data = data_subset_testing)

# Extract R-Squared value

rsq_bmismoker <- data_subset_testing %>%
  dplyr::select(smoker,charges) %>%
  bind_cols(charge_predictions_bmismoker) %>%
# Extract R-Squared value
  rsq(truth = charges, estimate = .pred) %>%
# Add identifier and keep only relevant columns
  mutate(dataset = "Test (Smoker + BMI)") %>%
  dplyr::select(dataset,.estimate)
```

Compare Testing Model for Healthcare Costs to alternatives using R-Squared Method

Now that we've extracted the R-Squared value from the alternative models, we'll do the same for the model fit to the testing and training data sets for Age+BMI+Smoker.

The R-Squared value measures the degree of linearity for the regression line. The straight the line, the stronger the relationship, with a minimum of 0 (no relationship) and maximum of 1 (perfectly related). The R-Squared value can also be used to explain how much of the differences in the predicted value (variance) is

explained by the model. For example, using our data, if the R-Squared is 50%, then 50% of the differences in healthcare costs can be explained by our model.

What is considered a good R-Squared value will vary by field and is best determined via discussion with stakeholders.

Now, let's take a look at our model to see how well it performs.

Combine model results

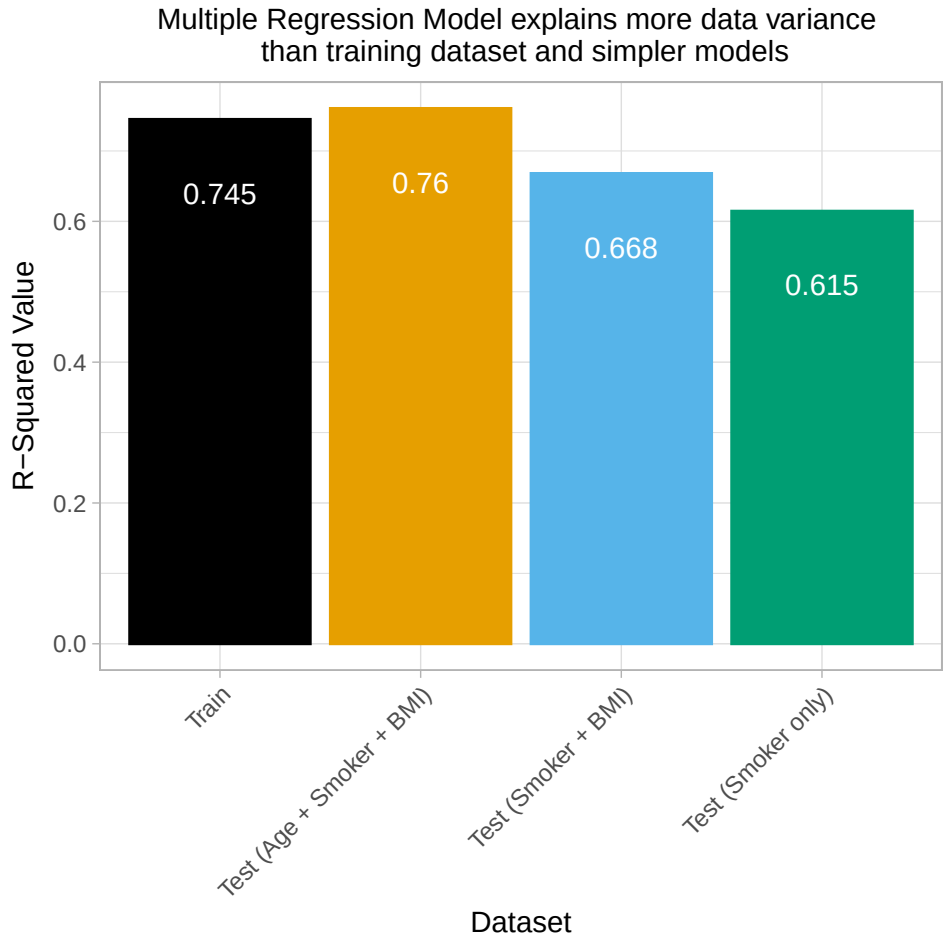
We'll first extract the R-Squared value from the training and testing data sets and merge these values with the R-Squared values generated from the alternative models. Then, we will compared the magnitude of each value to see which model performs the best.

Ideally, the Age+BMI+Smoker model will perform the best and have the highest R-Squared value, as predicted.

```
# Extract R-Squared value from training data
rsq_train <- charge_test_results_train %>%
  rsq(truth = charges, estimate = .pred) %>%
  mutate(dataset = "Train") %>%
  dplyr::select(dataset,.estimate)

# Extract R-Squared value from testing data
rsq_test <- charge_test_results_test %>%
  rsq(truth = charges, estimate = .pred) %>%
  mutate(dataset = "Test (Age + Smoker + BMI)") %>%
  dplyr::select(dataset,.estimate)

# Combind R-Squared data and give descriptive titles

rsq_combined <- rbind(rsq_train,rsq_test,rsq_smoker,rsq_bmismoker) %>%
  mutate(dataset = fct_relevel(dataset,
                               "Train",
                               "Test (Age + Smoker + BMI)",
                               "Test (Smoker + BMI)",
                               "Test (Smoker only)",))

# Initialize ggplot, comparing R-Squared value (y) across the models (x)

p <- ggplot(rsq_combined,(aes(dataset,.estimate,fill=dataset,color=dataset))) +

# Use geom_bar to create a bar plot with position = "dodge" to avoid stacking
  geom_bar(stat="identity",position="dodge") +

# Add data labels to each bar representing the R-Squared value
  geom_text(size=rel(4.0),
            vjust=4,color="white",
            aes(label=round(as.numeric(.estimate),3))) +

# Provide axes and plot titles
  xlab("Dataset") +
  ylab(expression("R-Squared Value")) +
  ggtitle("Multiple Regression Model explains more data variance
          \n than training dataset and simpler models") +
```

```
# Use a light theme and colorblind coloring for accessibility
  theme_light() +
  scale_color_colorblind() +
  scale_fill_colorblind() +

# Adjust axes labels to 45 degree angle and justify them to axis ticks
  theme(axis.text.x=element_text(size=rel(1.0),
                                 angle = 45,
                                 hjust = 1),
        plot.title = element_text(size=rel(1.0),
                                  hjust = 0.5),

# Remove legend
        legend.position = "none")

# Return plot

p
```

Plot results



It would appear that the model that includes Age+BMI+Smoker (Orange Bar) works the best, as it explains 76% of the differences in cost, which is higher than smoker status alone or Smoker+BMI.

Discussion

This concludes the coding sample of the project, and can be skipped unless you would like to see what else I would do to improve this analysis.

Improving the model

This was a fairly easy model to construct, as Kaggle includes data sets that have known predictor variables within the data to assist with model development. If the data were weaker, or if we wanted to improve the model anyways, we could use the following methods:

Collect more data to increase the number of variables or observations for each variable

This is the most difficult method, as it would require the addition of new data.

Normalize the data

I did not perform normalization on these data to save space, but normalized data would be more comparable across conditions and may increase model performance.

Add regularization to the model

As with normalization, this step in model construction was skipped, but it is possible that this model may be very effective with the current data, but be less applicable to future data. Regularization is a process by which we can minimize the impact of "noise" in the current data, making it more applicable to data acquired in the future.

Minimize the effect of small sample sizes

Although there are many observations in the insurance data set used for this analysis, there are methods to decrease the effect that a smaller sample size would have on the model. This is part of the regularization step explained in the above.

Conclusions

Using this workflow, we were successfully able to generate a statistical regression model that explains 76% of the differences in Healthcare costs among patients using Age, Body Mass Index (BMI), and whether they have a history of smoking or not. This model performed better than looking at smoker status alone or smoker status + BMI.

Depending on the needs of stakeholders, this model could be used as is or could be further improved to account for more of the differences in the data. We could increase the amount of data that is available to the model by collecting more observations or add more predictor variables to the data set. We could normalize the data to make the conditions more comparable to each other. Lastly, we could add regularization to reduce the negative impact that the current data set may have on the applicability of the model to additions acquired in the future.

Correspondence

Please address any questions to Myron Keith Gibert Jr at mkgibertjr@msn.com. Code for this project is stored in a GitHub repository.