Capstone Project

Tim Harvey

5/30/2017

The goal of this project is to determine how the local uninsured rate influences the readmission rate of hospitals and whether an increase in any of the HCAHPS linear scores has an effect.

The audience for this analysis is both hospital administrators and those that have an interest in insurance policy. If an administrator finds the local population to have a high uninsured rate, the ability to mitigate this with additional efforts targeted toward the most predictive HCAHPS scores to lower readmission rates. The readmission rate is a key indicator of quality of care and healthcare facilities are penalized by Medicare for high readmission rates. These penalties result in reimbursements being withheld and in October of 2016 these penalties were increased by 20 percent. To increase the quality of patient outcomes, advocates for increased insurance coverage may be interested in the analysis as evidence that increases in insurance coverage lower readmission rates. Lower readmission rates have the effect of lowering overall healthcare costs.

The data used for this analysis will be the Small Area Health Insurance Estimates (SAHIE) by the US Census Bureau (https://www.census.gov/did/www/sahie/data/20082014/index.html), and the hospital-level HCAHPS and readmission data available from Medicare (https://data.medicare.gov/data/hospital-compare).

The first step is using the SAHIE data to determine the mean uninsured rate for each county in the United States. Once the mean uninsured rate of the population served in each county is determined it can then be correlated to the mean readmission rate for hospitals in that county. If there is a correlation, then the results of the HCAHPS survey can be used to determine if there is a correlation between discharge counseling and readmission rate among hospitals that have similar rates of uninsured patients.

The key pieces of data in the Medicare data sets are the state and county, the readmission rate for each facility, and the HCAHPS scores for the metric associated with patient education. In the SAHIE data the key data fields are the state and county as well as the estimated uninsured rate in each county.

One of the limits of the data SAHIE data is that is the uninsured rate is an estimate. Ideally, this data would be a more exact metric, but SAHIE estimates are an accepted measure of the uninsured rate. Another limitation of the data is the three data sets do not measure the same period. The SAHIE data follows calendar years, January – December, whereas the period measured by the Medicare readmission data is July –

June, and the HCAHPS date is April - March. This is addressed by using the HCAHPS data time period, and weighting the time periods of the other two data sets.

Another issue with the data is the granularity of the uninsured rate. The uninsured rate is aggregated at the county level while the data for readmission and HCAHPS scores are collected at the hospital level. Any correlation would be more meaningful if all data was at the hospital level, avoiding the need to aggregate hospital-level metrics at the county level and yielding less accurate results. Additionally, potential users of the analysis would be handicapped by the aggregation, as many of the changes that could affect the readmission rate would take place at the hospital level. A missing element of the data set that would be useful would be bins for age ranges. As it is, there is no way to take age into consideration and conclusions can't be drawn regarding the impacts on various age groups.

The first step in munging the SAHIE data is removing the explanatory information at the top of the file, stripping the trailing white space, filtering out the data that covers specific populations, rather than the population as a whole, and selecting only the columns that are necessary to either further cleanse the data or for the final analysis. In this case, the year will be used to determine the final metric weighting, the state and county will be used to join the other data sets, and the PCTELIG column is the uninsured rate for the county population.

```
sahie_2014_clean <- read.csv(file.path("Data/", "sahie_2014.csv"), stringsAsFactors = FALSE,
    sep = ",", skip = 79, strip.white = TRUE, na.strings = c("")) %>% filter(agecat ==
    0, racecat == 0, sexcat == 0, iprcat == 0, county_name != "NA") %>% select(year,
    state_name, county_name, PCTELIG)
```

```
sahie_2015_clean <- read.csv(file.path("Data/", "sahie_2015.csv"), stringsAsFactors = FALSE,
    sep = ",", skip = 79, strip.white = TRUE, na.strings = c("")) %>% filter(agecat ==
    0, racecat == 0, sexcat == 0, iprcat == 0, county_name != "NA") %>% select(year,
    state_name, county_name, as.numeric(PCTELIG))
```

The next step is formatting the data in each column. The uninsured rate needs to be converted to a numeric data type, and the state and county names do not match those in the other two sets. The county name needs to be capitalized and the word "County" and the word "Parish" needed to be removed from the name. The state name is given as the full name, for example, "Louisiana" and needs to be converted to the abbreviation "LA".

```
sahie_2014_clean$PCTELIG <- as.numeric(sahie_2014_clean$PCTELIG)
sahie_2014_clean$state_name <- as.factor(state.abb[match(sahie_2014_clean$state_name,
    state.name)])
sahie_2014_clean$county_name <- as.factor(toupper(sahie_2014_clean$county_name))
sahie_2014_clean$county_name <- as.factor(gsub(" COUNTY", "", sahie_2014_clean$county_name))
sahie_2014_clean$county_name <- as.factor(gsub(" PARISH", "", sahie_2014_clean$county_name))
```

```
sahie_2015_clean$PCTELIG <- as.numeric(sahie_2015_clean$PCTELIG)
sahie_2015_clean$state_name <- as.factor(state.abb[match(sahie_2015_clean$state_name,
    state.name)])
sahie_2015_clean$county_name <- as.factor(toupper(sahie_2015_clean$county_name))
sahie_2015_clean$county_name <- as.factor(gsub(" COUNTY", "", sahie_2015_clean$county_name))
sahie_2015_clean$county_name <- as.factor(gsub(" PARISH", "", sahie_2015_clean$county_name))
```

The final step in cleansing the SAHIE data is combining the two years of data, with appropriate weighting, and averaging the score for all records within a county and state. The final version of the data set is completed by changing the columns headers for the county and state name to match those in the other two data sets, as well as changing the "PCTELIG" column name to the more descriptive "combined_sahie_metric".

```
sahie_final <- inner_join(sahie_2014_clean, sahie_2015_clean, by = c(state_name = "state_name",
    county_name = "county_name")) %>% mutate(combined_sahie_metric = (PCTELIG.x *
    0.75) + (PCTELIG.y * 0.25)) %>% select(state_name, county_name, PCTELIG.x, PCTELIG.y,
    combined_sahie_metric) %>% group_by(State = state_name, County.Name = as.factor(county_name)) %>%
    summarise(county_unins_average = mean(combined_sahie_metric))
```

The HCAHPS data is the most straightforward to clean as it is the base timeline, so there is only one file. The file is read, eliminating white space and formatting NA values. The required fields are then selected and the spread function is applied to make each HCAHPS measure a field of it's own.

```
hcahps_clean <- read.csv(file.path("Data", "HCAHPS_Hospital_201404-201503.csv"),
    stringsAsFactors = FALSE, sep = ",", strip.white = TRUE, na.strings = c("", "Not Available",
        "Not Applicable", "FEWER THAN 50")) %>% select(Provider.ID, State, County.Name,
    HCAHPS.Measure.ID, HCAHPS.Linear.Mean.Value) %>% spread(HCAHPS.Measure.ID, HCAHPS.Linear.Mean.Value
```

The state and county names are converted to factors.

```
hcahps_clean$State <- as.factor(hcahps_clean$State)
hcahps_clean$County.Name <- as.factor(hcahps_clean$County.Name)
```

The final HCAHPS data is generated by selecting the state name, county name, and the fields of the HCAHPS scores. The data is then across county and state, using a yielding a mean score for each county for each of the mean scores.

```
hcahps_select_target <- c("H_CLEAN_LINEAR_SCORE", "H_COMP_1_LINEAR_SCORE", "H_COMP_2_LINEAR_SCORE",
    "H_COMP_3_LINEAR_SCORE", "H_COMP_4_LINEAR_SCORE", "H_COMP_5_LINEAR_SCORE", "H_COMP_6_LINEAR_SCORE",
    "H_COMP_7_LINEAR_SCORE", "H_HSP_RATING_LINEAR_SCORE", "H_QUIET_LINEAR_SCORE",
    "H_RECMND_LINEAR_SCORE")
hcahps_final <- hcahps_clean %>% group_by(State, County.Name) %>% summarise_each(funs(mean(.,
    na.rm = TRUE)), one_of(hcahps_select_target))
```

The munging of the readmission data is very similar to the HCAHPS data, as they are from the same source and the same formatting conventions are used. The notable exception is there are two files that must be weighted to derive the final metric. The file is read eliminating white space and formatting NA values and the unnecessary scores and missing data are filtered out.

```
readmit_201307_201406_clean <- read.csv(file.path("Data", "Readmissions_and_Deaths 201307-201406_Hospita
    stringsAsFactors = FALSE, sep = ",", strip.white = TRUE, na.strings = c("", "Not Available")) %>%
    filter(Measure.ID == "READM_30_HOSP_WIDE", County.Name != "NA", Score != "NA") %>%
    select(Measure.Start.Date, Measure.End.Date, State, County.Name, Measure.ID,
        Score)
```

```
readmit_201407_201506_clean <- read.csv(file.path("Data", "Readmissions_and_Deaths 201407-201506_Hospita
    stringsAsFactors = FALSE, sep = ",", strip.white = TRUE, na.strings = c("", "Not Available")) %>%
    filter(Measure.ID == "READM_30_HOSP_WIDE", County.Name != "NA", Score != "NA") %>%
    select(Measure.Start.Date, Measure.End.Date, State, County.Name, Measure.ID,
        Score)
```

The dates are formatted in the same way as the HCAHPS file, so the same gsub is used. The state and county names are then converted to factors.

```
readmit_201307_201406_clean$Measure.Start.Date <- as.Date(gsub("7/1/13", "07/01/2013",
    readmit_201307_201406_clean$Measure.Start.Date), "%m/%d/%Y")
readmit_201307_201406_clean$Measure.End.Date <- as.Date(gsub("6/30/14", "06/30/2014",
```

```
    readmit_201307_201406_clean$Measure.End.Date), "%m/%d/%Y")
readmit_201307_201406_clean$State <- as.factor(readmit_201307_201406_clean$State)
readmit_201307_201406_clean$County.Name <- as.factor(readmit_201307_201406_clean$County.Name)
```

```
readmit_201407_201506_clean$Measure.Start.Date <- as.Date(gsub("7/1/14", "07/01/2014",
    readmit_201407_201506_clean$Measure.Start.Date), "%m/%d/%Y")
readmit_201407_201506_clean$Measure.End.Date <- as.Date(gsub("6/30/15", "06/30/2015",
    readmit_201407_201506_clean$Measure.End.Date), "%m/%d/%Y")
readmit_201407_201506_clean$State <- as.factor(readmit_201407_201506_clean$State)
readmit_201407_201506_clean$County.Name <- as.factor(readmit_201407_201506_clean$County.Name)
```

The final readmission data is derived by joining the two yearly files by county and state. The readmission rates are then weighted and combined into a final metric. The final columns are State, County.Name, the individual yearly scores, and the final combined metric. The mean of the combined metric per county is derived, yielding the final readmission file.

```
readmit_final <- inner_join(readmit_201307_201406_clean, readmit_201407_201506_clean,
    by = c(State = "State", County.Name = "County.Name")) %>% mutate(combined_readmit_metric = (Score.x
    0.25) + (Score.y * 0.75)) %>% select(State, County.Name, Score.x, Score.y, combined_readmit_metric)
    group_by(State = as.factor(State), County.Name = as.factor(County.Name)) %>%
    summarise(county_readmit_average = mean(combined_readmit_metric))
```

The final data set on which the analysis will be preformed is derived by using nested joins and the state and county names.

```
data_final <- inner_join(readmit_final, hcahps_final, by = c(State = "State", County.Name = "County.Nam
    inner_join(., sahie_final, by = c(State = "State", County.Name = "County.Name"))
str(data_final)
```

```
## Classes 'grouped_df', 'tbl_df', 'tbl' and 'data.frame':  2280 obs. of  15 variables:
##  $ State                    : chr  "AL" "AL" "AL" "AL" ...
##  $ County.Name              : chr  "AUTAUGA" "BALDWIN" "BARBOUR" "BIBB" ...
##  $ county_readmit_average   : num  14.6 15.7 15.3 15.3 15.4 ...
##  $ H_CLEAN_LINEAR_SCORE     : num  87 88 89 NaN 86 ...
##  $ H_COMP_1_LINEAR_SCORE    : num  92 93 92 NaN 93 ...
##  $ H_COMP_2_LINEAR_SCORE    : num  93 93 94 NaN 94 ...
```

```
##  $ H_COMP_3_LINEAR_SCORE    : num  84 87 83 NaN 91 NaN 89 80 NaN NaN ...
##  $ H_COMP_4_LINEAR_SCORE    : num  89 89.3 86 NaN 94 ...
##  $ H_COMP_5_LINEAR_SCORE    : num  82 81 78 NaN 80 NaN 81 71 NaN NaN ...
##  $ H_COMP_6_LINEAR_SCORE    : num  89 88 84 NaN 82 ...
##  $ H_COMP_7_LINEAR_SCORE    : num  82 82 81 NaN 80 ...
##  $ H_HSP_RATING_LINEAR_SCORE: num  91 91 88 NaN 92 ...
##  $ H_QUIET_LINEAR_SCORE     : num  88 89 89 NaN 88 ...
##  $ H_RECMND_LINEAR_SCORE    : num  90 90 85 NaN 89 NaN 88 87 NaN NaN ...
##  $ county_unins_average     : num  10.6 15 14.8 13.2 15.9 ...
##  - attr(*, "vars")=List of 1
##   ..$ : symbol State
```

This analysis shows there is almost no correlation between the uninsured rate and the readmission rate in each county.

```
analyze_data <- data_final[c(3, 15)]
lm1 <- lm(county_readmit_average ~ county_unins_average, data = analyze_data)
summary(lm1)
```

```
##
## Call:
## lm(formula = county_readmit_average ~ county_unins_average, data = analyze_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0802 -0.3497 -0.0505  0.3087  3.6974
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          15.338042   0.036441 420.896   <2e-16 ***
## county_unins_average  0.008231   0.002574   3.198   0.0014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.5939 on 2278 degrees of freedom
## Multiple R-squared:  0.00447,    Adjusted R-squared:  0.004033
## F-statistic: 10.23 on 1 and 2278 DF,  p-value: 0.001401
```

```
str(data_final[c(3, 15)])
```

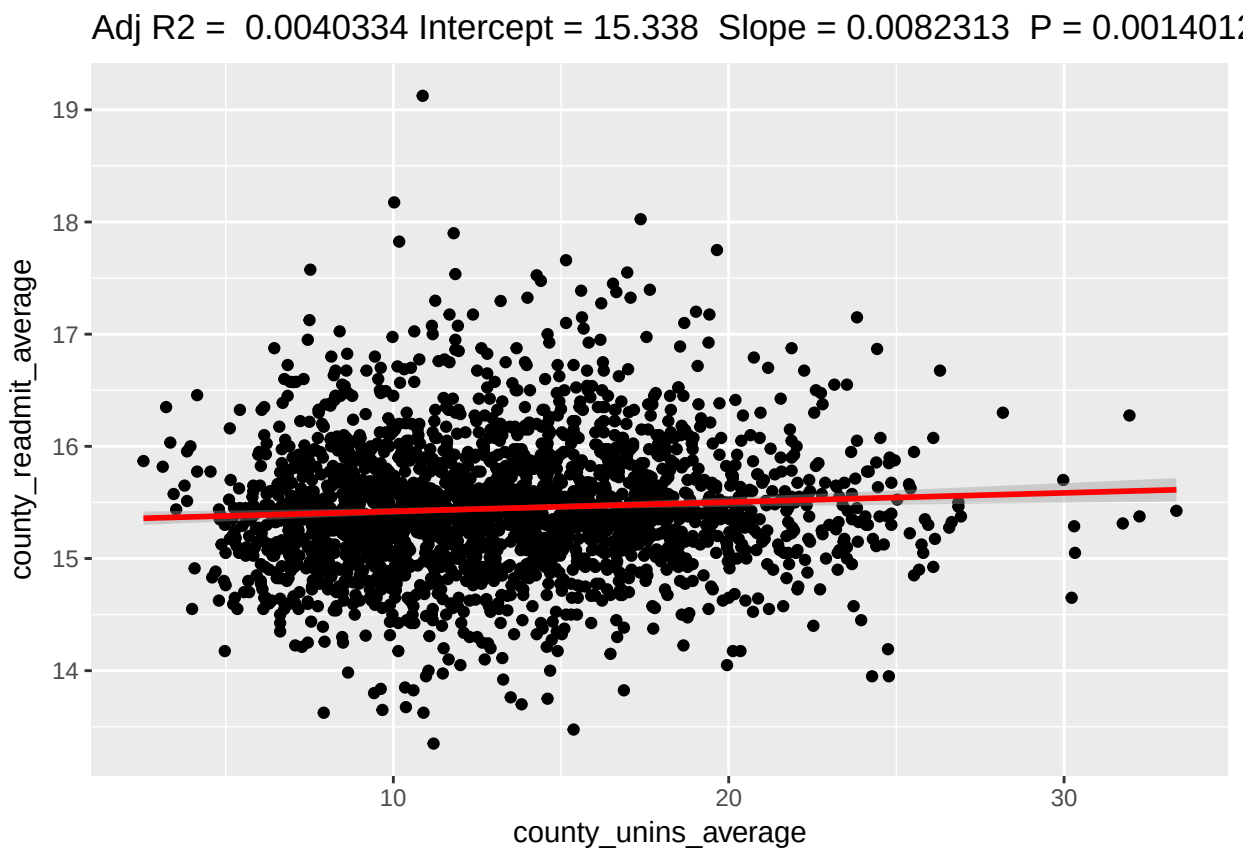
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2280 obs. of  2 variables:
##  $ county_readmit_average: num  14.6 15.7 15.3 15.3 15.4 ...
##  $ county_unins_average  : num  10.6 15 14.8 13.2 15.9 ...
```

```
ggplotRegression <- function(fit) {

    require(ggplot2)

    ggplot(fit$model, aes_string(x = names(fit$model)[2], y = names(fit$model)[1])) +
        geom_point() + stat_smooth(method = "lm", col = "red") + labs(title = paste("Adj R2 = ",
        signif(summary(fit)$adj.r.squared, 5), "Intercept =", signif(fit$coef[[1]],
            5), " Slope =", signif(fit$coef[[2]], 5), " P =", signif(summary(fit)$coef[2,
            4], 5)))
}
```

```
ggplotRegression(lm1)
```



The highest adjusted R squared value for a single attribute, when aggregated at the county level is 0.08.

```
for (i in 4:15) {
    analyze_data <- data_final[c(3, i)]
    lm1 <- lm(county_readmit_average ~ ., data = analyze_data)
    print(colnames(data_final[i]))
    print(summary(lm1)$adj.r.squared)
}
```

```
## [1] "H_CLEAN_LINEAR_SCORE"
## [1] 0.02409226
## [1] "H_COMP_1_LINEAR_SCORE"
## [1] 0.03833288
## [1] "H_COMP_2_LINEAR_SCORE"
## [1] 0.01506094
## [1] "H_COMP_3_LINEAR_SCORE"
## [1] 0.062421
```

```
## [1] "H_COMP_4_LINEAR_SCORE"
## [1] 0.02963111
## [1] "H_COMP_5_LINEAR_SCORE"
## [1] 0.03796599
## [1] "H_COMP_6_LINEAR_SCORE"
## [1] 0.0817606
## [1] "H_COMP_7_LINEAR_SCORE"
## [1] 0.06167917
## [1] "H_HSP_RATING_LINEAR_SCORE"
## [1] 0.05313558
## [1] "H_QUIET_LINEAR_SCORE"
## [1] 0.006334165
## [1] "H_RECMND_LINEAR_SCORE"
## [1] 0.05960038
## [1] "county_unins_average"
## [1] 0.004033388
```

Because of the low adjusted R squared value for the county uninsured rate it can be removed as a contributing factor and the HCAHPS scores and the readmission rate can be evaluated at the hospital level. The data cleansing is largely the same with the exception of discarding the Provider ID and averaging the results across the County and State pairs.

Readmission data cleansing

```
readmit_201307_201406_clean <- read.csv(file.path("Data", "Readmissions_and_Deaths 201307-201406_Hospita
    stringsAsFactors = FALSE, sep = ",", strip.white = TRUE, na.strings = c("", "Not Available")) %>%
    filter(Measure.ID == "READM_30_HOSP_WIDE", Score != "NA") %>% select(Provider.ID,
    Score)

readmit_201307_201406_clean$Provider.ID <- as.factor(readmit_201307_201406_clean$Provider.ID)
str(readmit_201307_201406_clean)
```

```
## 'data.frame':    4448 obs. of  2 variables:
##  $ Provider.ID: Factor w/ 4448 levels "100001","100002",..: 7 33 41 47 54 70 75 95 110 116 ...
##  $ Score      : num  14.8 15.2 15.3 15.4 14.6 14.5 15.6 14.5 15.8 14.3 ...
```

```
readmit_201407_201506_clean <- read.csv(file.path("Data", "Readmissions_and_Deaths 201407-201506_Hospita
    stringsAsFactors = FALSE, sep = ",", strip.white = TRUE, na.strings = c("", "Not Available")) %>%
    filter(Measure.ID == "READM_30_HOSP_WIDE", Score != "NA") %>% select(Provider.ID,
    Score)

readmit_201407_201506_clean$Provider.ID <- as.factor(readmit_201407_201506_clean$Provider.ID)
str(readmit_201407_201506_clean)
```

```
## 'data.frame':    4394 obs. of  2 variables:
##  $ Provider.ID: Factor w/ 4394 levels "10001","10005",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Score      : num  15.4 14.9 15.4 16.6 15.7 15.3 15.3 14.7 15.9 14.8 ...
```

```
readmit_final <- inner_join(readmit_201307_201406_clean, readmit_201407_201506_clean,
    by = c(Provider.ID = "Provider.ID")) %>% mutate(combined_readmit_metric = (Score.x *
    0.25) + (Score.y * 0.75)) %>% select(Provider.ID, combined_readmit_metric)
```

```
## Warning in inner_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## factors with different levels, coercing to character vector
```

```
str(readmit_final)
```

```
## 'data.frame':    4327 obs. of  2 variables:
##  $ Provider.ID            : chr  "10001" "10005" "10006" "10007" ...
##  $ combined_readmit_metric: num  15.2 15 15.4 16.3 15.4 ...
```

HCAHPS data cleansing

```
hcahps_select_target <- c("H_CLEAN_LINEAR_SCORE", "H_COMP_1_LINEAR_SCORE", "H_COMP_2_LINEAR_SCORE",
    "H_COMP_3_LINEAR_SCORE", "H_COMP_4_LINEAR_SCORE", "H_COMP_5_LINEAR_SCORE", "H_COMP_6_LINEAR_SCORE",
    "H_COMP_7_LINEAR_SCORE", "H_HSP_RATING_LINEAR_SCORE", "H_QUIET_LINEAR_SCORE",
    "H_RECMND_LINEAR_SCORE")
hcahps_final <- read.csv(file.path("Data", "HCAHPS_Hospital_201404-201503.csv"),
    stringsAsFactors = FALSE, sep = ",", strip.white = TRUE, na.strings = c("", "Not Available",
        "Not Applicable", "FEWER THAN 50")) %>% filter(HCAHPS.Measure.ID %in% hcahps_select_target) %>%
    select(Provider.ID, HCAHPS.Measure.ID, HCAHPS.Linear.Mean.Value) %>% spread(HCAHPS.Measure.ID,
    HCAHPS.Linear.Mean.Value)
```

```
hcahps_final$Provider.ID <- as.factor(hcahps_final$Provider.ID)
```

Joining HCAHPS data and readmission data for final analysis set.

```
data_final <- inner_join(readmit_final, hcahps_final, by = c(Provider.ID = "Provider.ID"))
```

```
## Warning in inner_join_impl(x, y, by$x, by$y, suffix$x, suffix$y): joining
## character vector and factor, coercing into character vector
```

Adjusted R squared for all variables in data set.

```
analyze_data <- data_final[c(2:13)]
lm1 <- lm(combined_readmit_metric ~ ., data = analyze_data)
summary(lm1)
```

```
##
## Call:
## lm(formula = combined_readmit_metric ~ ., data = analyze_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8850 -0.4804 -0.0260  0.4440  3.3144
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                21.826797   0.749743  29.112  < 2e-16 ***
## H_CLEAN_LINEAR_SCORE       -0.005656   0.005258  -1.076 0.282158
## H_COMP_1_LINEAR_SCORE       0.080147   0.014810   5.412 6.68e-08 ***
## H_COMP_2_LINEAR_SCORE      -0.003523   0.009152  -0.385 0.700306
## H_COMP_3_LINEAR_SCORE      -0.048224   0.006891  -6.998 3.11e-12 ***
## H_COMP_4_LINEAR_SCORE      -0.018382   0.010570  -1.739 0.082122 .
## H_COMP_5_LINEAR_SCORE       0.015086   0.006044   2.496 0.012605 *
## H_COMP_6_LINEAR_SCORE      -0.047022   0.005312  -8.853  < 2e-16 ***
## H_COMP_7_LINEAR_SCORE      -0.024302   0.011631  -2.089 0.036739 *
## H_HSP_RATING_LINEAR_SCORE   0.011206   0.014844   0.755 0.450381
## H_QUIET_LINEAR_SCORE       -0.002090   0.003752  -0.557 0.577583
```

```
## H_RECMND_LINEAR_SCORE     -0.034882   0.009533  -3.659 0.000257 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7793 on 3443 degrees of freedom
##   (872 observations deleted due to missingness)
## Multiple R-squared:  0.1572, Adjusted R-squared:  0.1545
## F-statistic: 58.38 on 11 and 3443 DF,  p-value: < 2.2e-16
```

Splitting the data for testing. 50% will be used for the training set and 50% will be used as the test set.

```
indexes <- sample(1:nrow(data_final), size = 0.5 * nrow(data_final))


data_final_test <- data_final[-indexes, ]
data_final_train <- data_final[indexes, ]
```

Run summary to determine adjusted R squared for all variables.

```
analyze_data_train <- data_final_train[c(2:13)]
lm2 <- lm(combined_readmit_metric ~ ., data = data_final_train[c(2:13)])
summary(lm2)
```

```
##
## Call:
## lm(formula = combined_readmit_metric ~ ., data = data_final_train[c(2:13)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8277 -0.4859 -0.0398  0.4525  3.3412
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               20.669125   1.032891  20.011  < 2e-16 ***
## H_CLEAN_LINEAR_SCORE      -0.005554   0.007440  -0.747 0.455463
## H_COMP_1_LINEAR_SCORE      0.085261   0.021207   4.020 6.06e-05 ***
## H_COMP_2_LINEAR_SCORE     -0.003344   0.012818  -0.261 0.794241
```

```
## H_COMP_3_LINEAR_SCORE     -0.045461   0.009695  -4.689 2.96e-06 ***
## H_COMP_4_LINEAR_SCORE     -0.004108   0.015189  -0.270 0.786819
## H_COMP_5_LINEAR_SCORE      0.013646   0.008786   1.553 0.120541
## H_COMP_6_LINEAR_SCORE     -0.042755   0.007510  -5.693 1.46e-08 ***
## H_COMP_7_LINEAR_SCORE     -0.023736   0.016349  -1.452 0.146720
## H_HSP_RATING_LINEAR_SCORE  0.005965   0.020691   0.288 0.773144
## H_QUIET_LINEAR_SCORE       0.001707   0.005288   0.323 0.746898
## H_RECMND_LINEAR_SCORE     -0.046033   0.013256  -3.473 0.000528 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7889 on 1732 degrees of freedom
##   (419 observations deleted due to missingness)
## Multiple R-squared:  0.149,  Adjusted R-squared:  0.1436
## F-statistic: 27.58 on 11 and 1732 DF,  p-value: < 2.2e-16
```

Remove variables with lower significance in attempt to increase adjusted R squared.

```
lm2 <- lm(combined_readmit_metric ~ ., data = data_final_train[c(2, 4, 6, 8, 9, 10,
    13)])
summary(lm2)
```

```
##
## Call:
## lm(formula = combined_readmit_metric ~ ., data = data_final_train[c(2,
##     4, 6, 8, 9, 10, 13)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8433 -0.4897 -0.0403  0.4512  3.3473
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           20.516763   0.881600  23.272  < 2e-16 ***
```

```
## H_COMP_1_LINEAR_SCORE  0.080969   0.018846   4.296 1.83e-05 ***
## H_COMP_3_LINEAR_SCORE -0.046570   0.009135  -5.098 3.80e-07 ***
## H_COMP_5_LINEAR_SCORE  0.013106   0.008375   1.565    0.118
## H_COMP_6_LINEAR_SCORE -0.041979   0.007209  -5.823 6.86e-09 ***
## H_COMP_7_LINEAR_SCORE -0.025181   0.015859  -1.588    0.113
## H_RECMND_LINEAR_SCORE -0.043196   0.008085  -5.342 1.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7879 on 1737 degrees of freedom
##   (419 observations deleted due to missingness)
## Multiple R-squared:  0.1486, Adjusted R-squared:  0.1457
## F-statistic: 50.53 on 6 and 1737 DF,  p-value: < 2.2e-16
```

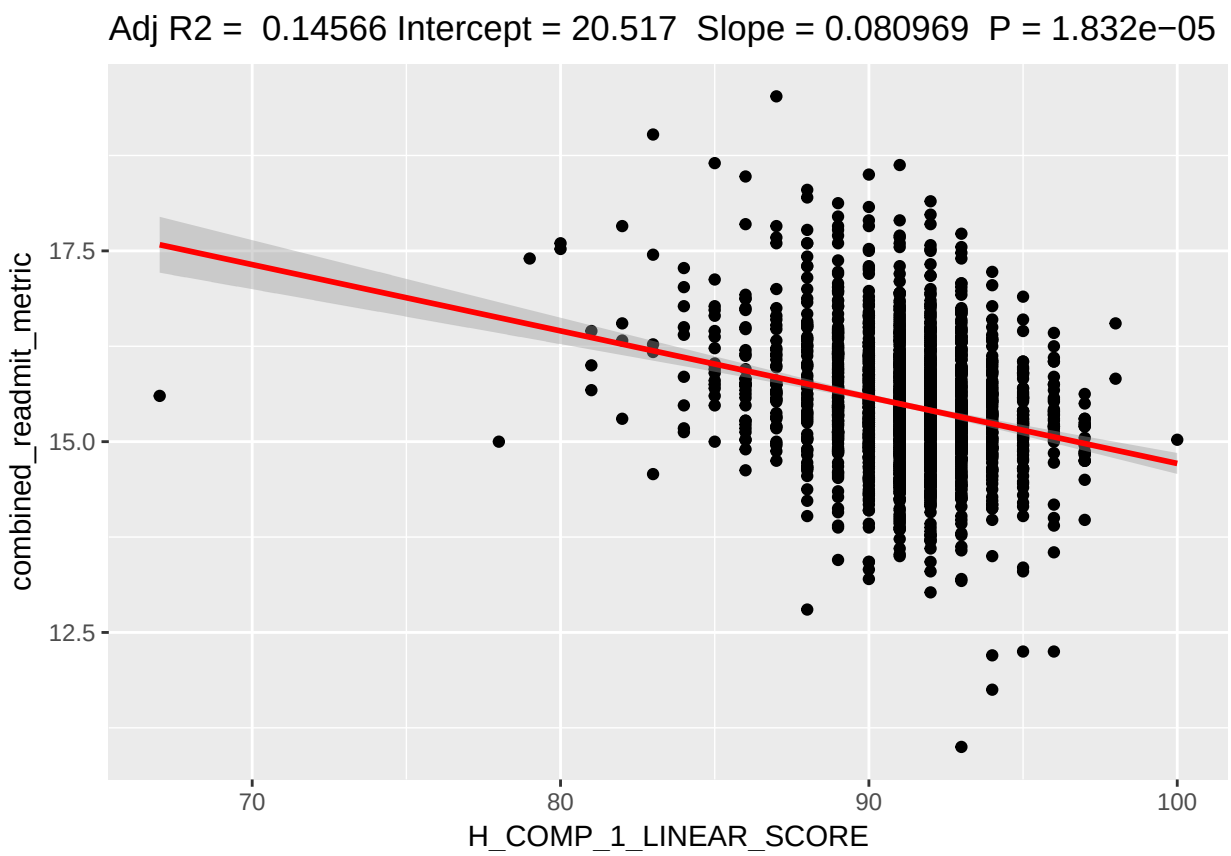
Create model

```
readmit_hcahps_model <- lm(combined_readmit_metric ~ H_COMP_1_LINEAR_SCORE + H_COMP_3_LINEAR_SCORE +
    H_COMP_5_LINEAR_SCORE + H_COMP_6_LINEAR_SCORE + H_COMP_7_LINEAR_SCORE + H_RECMND_LINEAR_SCORE,
    data = data_final_train)
summary(readmit_hcahps_model)
```

```
##
## Call:
## lm(formula = combined_readmit_metric ~ H_COMP_1_LINEAR_SCORE +
##     H_COMP_3_LINEAR_SCORE + H_COMP_5_LINEAR_SCORE + H_COMP_6_LINEAR_SCORE +
##     H_COMP_7_LINEAR_SCORE + H_RECMND_LINEAR_SCORE, data = data_final_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8433 -0.4897 -0.0403  0.4512  3.3473
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           20.516763   0.881600  23.272  < 2e-16 ***
```

```
## H_COMP_1_LINEAR_SCORE  0.080969   0.018846   4.296 1.83e-05 ***
## H_COMP_3_LINEAR_SCORE -0.046570   0.009135  -5.098 3.80e-07 ***
## H_COMP_5_LINEAR_SCORE  0.013106   0.008375   1.565    0.118
## H_COMP_6_LINEAR_SCORE -0.041979   0.007209  -5.823 6.86e-09 ***
## H_COMP_7_LINEAR_SCORE -0.025181   0.015859  -1.588    0.113
## H_RECMND_LINEAR_SCORE -0.043196   0.008085  -5.342 1.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7879 on 1737 degrees of freedom
##   (419 observations deleted due to missingness)
## Multiple R-squared:  0.1486, Adjusted R-squared:  0.1457
## F-statistic: 50.53 on 6 and 1737 DF,  p-value: < 2.2e-16
```

```
ggplotRegression(readmit_hcahps_model)
```

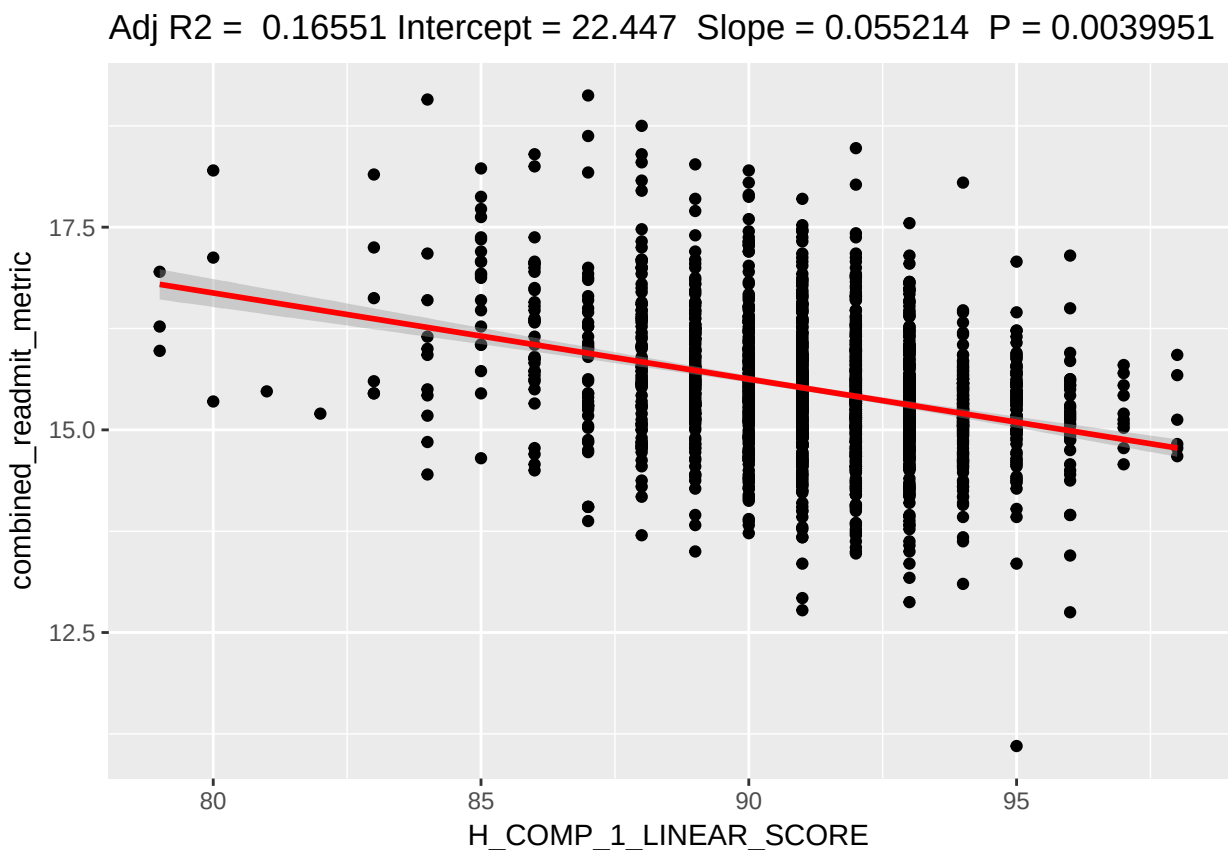


Run model with test data set

```r
readmit_hcahps_model_test <- lm(combined_readmit_metric ~ H_COMP_1_LINEAR_SCORE +
    H_COMP_3_LINEAR_SCORE + H_COMP_5_LINEAR_SCORE + H_COMP_6_LINEAR_SCORE + H_COMP_7_LINEAR_SCORE +
    H_RECMND_LINEAR_SCORE, data = data_final_test)
summary(readmit_hcahps_model_test)
```

```
##
## Call:
## lm(formula = combined_readmit_metric ~ H_COMP_1_LINEAR_SCORE +
##     H_COMP_3_LINEAR_SCORE + H_COMP_5_LINEAR_SCORE + H_COMP_6_LINEAR_SCORE +
##     H_COMP_7_LINEAR_SCORE + H_RECMND_LINEAR_SCORE, data = data_final_test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7475 -0.4621 -0.0114  0.4321  3.0804
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           22.447139   0.905137  24.800  < 2e-16 ***
## H_COMP_1_LINEAR_SCORE  0.055214   0.019155   2.882   0.0040 **
## H_COMP_3_LINEAR_SCORE -0.055409   0.009298  -5.959 3.07e-09 ***
## H_COMP_5_LINEAR_SCORE  0.014146   0.008114   1.743   0.0814 .
## H_COMP_6_LINEAR_SCORE -0.046792   0.007350  -6.366 2.48e-10 ***
## H_COMP_7_LINEAR_SCORE -0.035021   0.015982  -2.191   0.0286 *
## H_RECMND_LINEAR_SCORE -0.016736   0.008131  -2.058   0.0397 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7697 on 1704 degrees of freedom
##   (453 observations deleted due to missingness)
## Multiple R-squared:  0.1684, Adjusted R-squared:  0.1655
## F-statistic: 57.53 on 6 and 1704 DF,  p-value: < 2.2e-16
```

```
ggplotRegression(readmit_hcahps_model_test)
```



Clearly there is not a strong correlation between any one of the HCAHPS factors and readmission rate and only a slight correlation using the best combination of factors. That said, the analysis can still of use to those aiming to reduce readmission rate. With limited resources focus is best applied to improve those metrics that measure how well patients feel they are educated upon discharge (H_COMP_6_LINEAR_SCORE, H_COMP_7_LINEAR_SCORE) and the responsiveness of the staff (H_COMP_3_LINEAR_SCORE).

While the analysis of the HCAHPS data is not necessarily predictive, it does offer guidance as to which secondary factors focus may be best applied to improve patient outcomes.