

HKS SUP-135: Bias in Algorithms

Matt Khinda

4/14/2023

```
# Set random seed
HUID <- 41531460
set.seed(HUID)

# Store predictor variables
pred_vars <- colnames(health[,grep("^tm1", names(health))])

# Assign random number then flag for testing or training set (90-10 split)
health$random_number <- runif(length(health$patient_id))
health$train_flag <- ifelse(health$random_number<= 0.1, 1, 0)

# Create variable to exclude race
race <- c("tm1_dem_black")
exclude_race <- setdiff(pred_vars,race)
```

Question 1: Splitting Data into Training and Test Sets

```
# Subset to create training set
training <- subset(health, train_flag == 1)
test <- subset(health, train_flag == 0)
```

After dividing the full dataset randomly into a 10% training sample and 90% test sample, the resulting datasets contain 4989 observations and 43795 observations respectively.

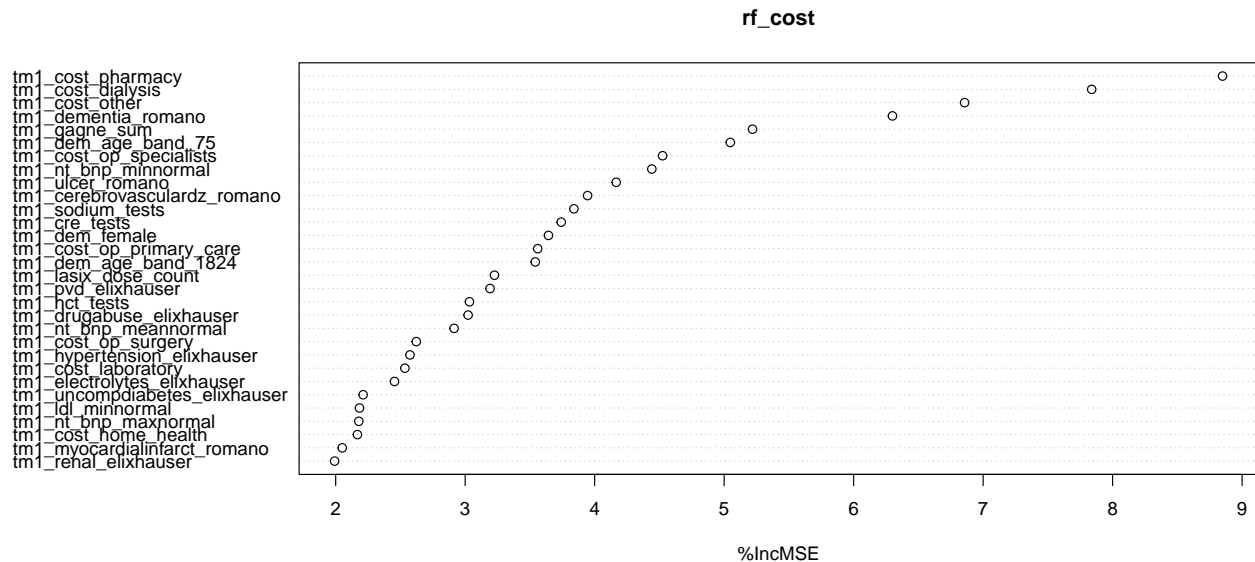
Question 2: Random Forests

```
rf_cost <- randomForest(reformulate(exclude_race, "cost_t"),
                        ntree=100,
                        mtry=149,
                        importance=TRUE,
                        data=training)

# Create predictions from model
y_test_predictions_rf_cost <- predict(rf_cost, newdata=test)
y_train_predictions_rf_cost <- predict(rf_cost, newdata=training)
```

```
# Plot variable importance
varImpPlot(rf_cost, type=1)
```

2a: Random Forest Using Cost Excluding Race

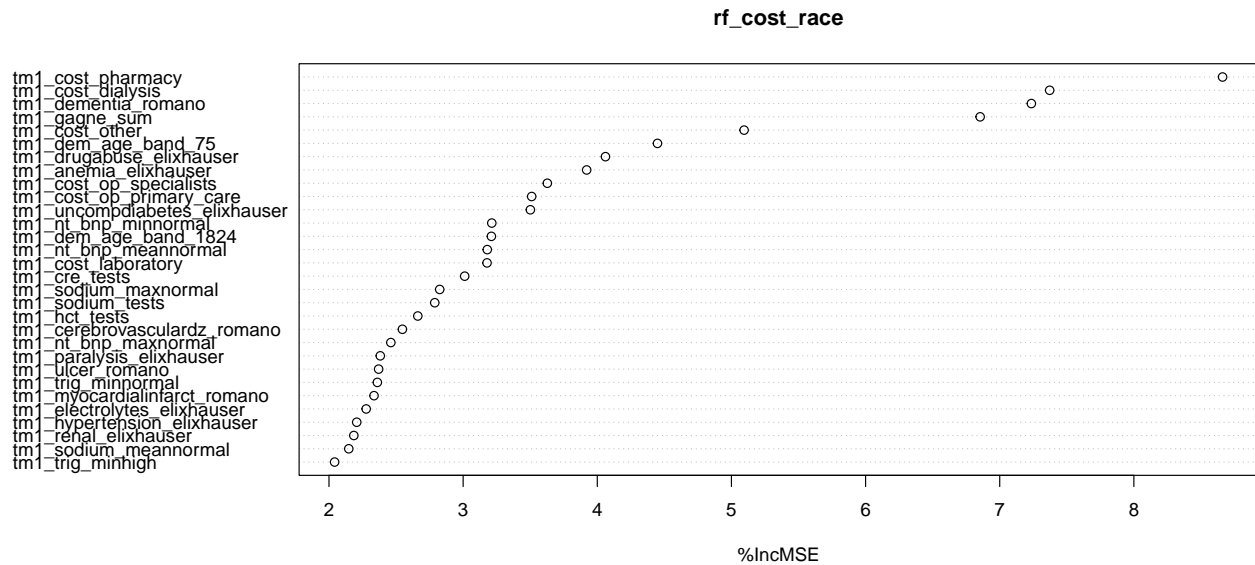


```
rf_cost_race <- randomForest(reformulate(pred_vars, "cost_t"),
                             ntree=100,
                             mtry=149,
                             importance=TRUE,
                             data=training)

# Create predictions from model
y_test_predictions_rf_cost_race <- predict(rf_cost_race, newdata=test)
y_train_predictions_rf_cost_race <- predict(rf_cost_race, newdata=training)

# Plot variable importance
varImpPlot(rf_cost_race, type=1)
```

2b: Random Forest Using Cost Including Race

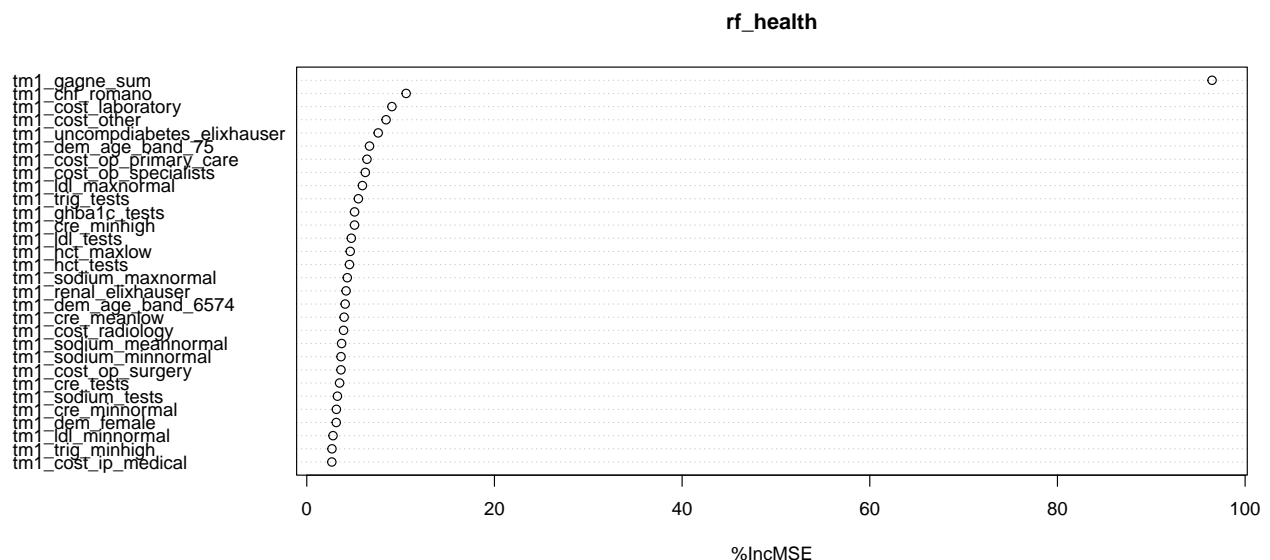


```
rf_health <- randomForest(reformulate(exclude_race, "gagne_sum_t"),
  ntree=100,
  mtry=149,
  importance=TRUE,
  data=training)

# Create predictions from model
y_test_predictions_rf_health <- predict(rf_health, newdata=test)
y_train_predictions_rf_health <- predict(rf_health, newdata=training)

# Plot variable importance
varImpPlot(rf_health, type=1)
```

2c: Random Forest Using Health Excluding Race



```

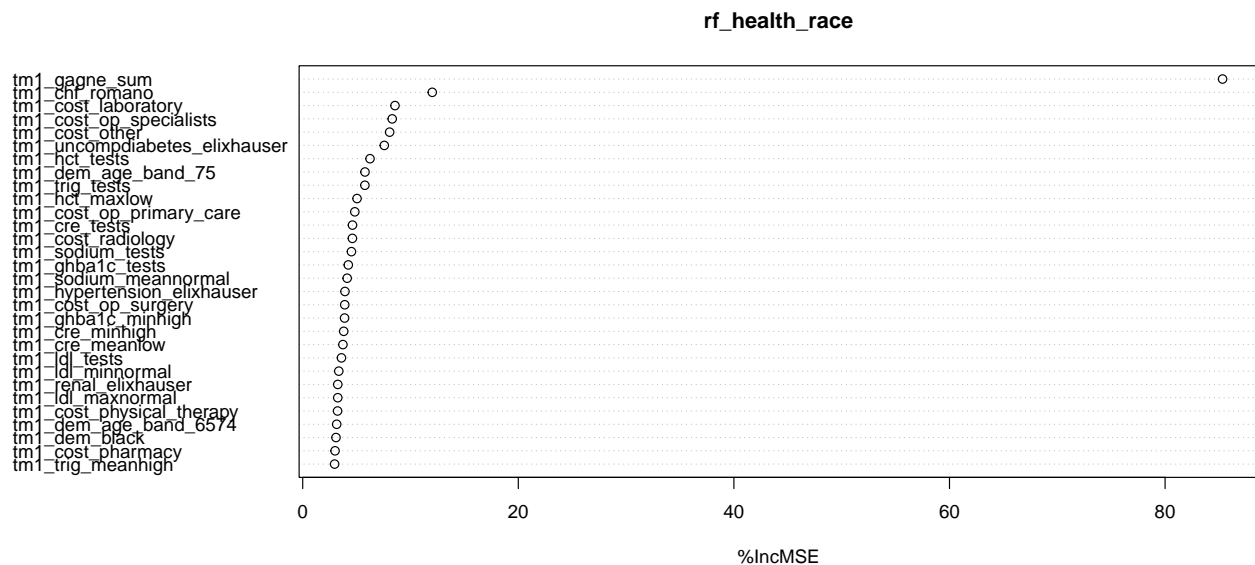
rf_health_race <- randomForest(reformulate(pred_vars, "gagne_sum_t"),
                               ntree=100,
                               mtry=149,
                               importance=TRUE,
                               data=training)

# Create predictions from model
y_test_predictions_rf_health_race <- predict(rf_health_race, newdata=test)
y_train_predictions_rf_health_race <- predict(rf_health_race, newdata=training)

# Plot variable importance
varImpPlot(rf_health_race, type=1)

```

2d: Random Forest Using Health Including Race



Question 3: Comparing RMSPE for Training Sample

```

# Create placeholder matrix
p <- 4
RMSPE <- matrix(0, p, 1)

## Model 1
RMSPE[1] <- sqrt(mean((training$cost_t -
                       y_train_predictions_rf_cost)^2, na.rm=TRUE))

## Model 2
RMSPE[2] <- sqrt(mean((training$cost_t -
                       y_train_predictions_rf_cost_race)^2, na.rm=TRUE))

## Model 3

```

```

RMSPE[3] <- sqrt(mean((training$gagne_sum_t -
                        y_train_predictions_rf_health)^2, na.rm=TRUE))

## Model 4
RMSPE[4] <- sqrt(mean((training$gagne_sum_t -
                        y_train_predictions_rf_health_race)^2, na.rm=TRUE))

# Create a data frame of results
data.frame(algorithm = c("Model 1 - Costs (excl. race) ",
                          "Model 2 - Costs (incl. race) ",
                          "Model 3 - Health (excl. race)",
                          "Model 4 - Health (incl. race)"),
            RMSPE)

```

```

##
## 1 Model 1 - Costs (excl. race) 6323.3312835
## 2 Model 2 - Costs (incl. race) 6344.5803796
## 3 Model 3 - Health (excl. race) 0.4607782
## 4 Model 4 - Health (incl. race) 0.4529860

```

Question 4: Comparing RMSPE for Test Sample

```

# Create placeholder matrix
p <- 4
RMSPE <- matrix(0, p, 1)

## Model 1
RMSPE[1] <- sqrt(mean((test$cost_t -
                        y_train_predictions_rf_cost)^2, na.rm=TRUE))

## Model 2
RMSPE[2] <- sqrt(mean((test$cost_t -
                        y_train_predictions_rf_cost_race)^2, na.rm=TRUE))

## Model 3
RMSPE[3] <- sqrt(mean((test$gagne_sum_t -
                        y_train_predictions_rf_health)^2, na.rm=TRUE))

## Model 4
RMSPE[4] <- sqrt(mean((test$gagne_sum_t -
                        y_train_predictions_rf_health_race)^2, na.rm=TRUE))

# Create a data frame of results
data.frame(algorithm = c("Model 1 - Costs (excl. race) ",
                          "Model 2 - Costs (incl. race) ",
                          "Model 3 - Health (excl. race)",
                          "Model 4 - Health (incl. race)"),
            RMSPE)

```

```

##
## 1 Model 1 - Costs (excl. race) 21252.402916

```

```
## 2 Model 2 - Costs (incl. race) 21245.978010
## 3 Model 3 - Health (excl. race) 2.618805
## 4 Model 4 - Health (incl. race) 2.621005
```

Question 5: Export Data

```
lab8_results <- test

lab8_results$y_test_preds_rf_cost <- y_test_predictions_rf_cost
lab8_results$y_test_preds_rf_cost_race <- y_test_predictions_rf_cost_race
lab8_results$y_test_preds_rf_health <- y_test_predictions_rf_health
lab8_results$y_test_preds_rf_health_race <- y_test_predictions_rf_health_race

write_dta(lab8_results, here("datasets", "Lab8_results.dta"))
```

Question 6: Percentile Rank Risk Scores

```
# assign predictions to test frame
test$y_test_predictions_rf_cost <- y_test_predictions_rf_cost
test$y_test_predictions_rf_cost_race <- y_test_predictions_rf_cost_race
test$y_test_predictions_rf_health <- y_test_predictions_rf_health
test$y_test_predictions_rf_health_race <- y_test_predictions_rf_health_race

# Define pct_rank function
pct_rank <- function(x){
  #Catch NAs
  r <- ifelse(is.na(x), NA, rank(x, ties.method = "average"))
  #return percentile rank
  100*r/max(r, na.rm = TRUE)
}

# Create percentile rank risk scores from predictions
test$risk_cost <- pct_rank(test$y_test_predictions_rf_cost)
test$risk_cost_race <- pct_rank(test$y_test_predictions_rf_cost_race)
test$risk_health <- pct_rank(test$y_test_predictions_rf_health)
test$risk_health_race <- pct_rank(test$y_test_predictions_rf_health_race)
```

Question 7: Eligibility

```
# Create eligibility indicator variables
test$mod1_eligibility <- ifelse(test$risk_cost > 55, 1, 0)
test$mod2_eligibility <- ifelse(test$risk_cost_race > 55, 1, 0)
test$mod3_eligibility <- ifelse(test$risk_health > 55, 1, 0)
test$mod4_eligibility <- ifelse(test$risk_health_race > 55, 1, 0)
```

7a: Eligibility Indicators

```

# Create placeholder matrix
p <- 4
black_eligibility <- matrix(0, p, 1)

# Calculate mean eligibility
black_eligibility[1] <- mean(subset(test, tm1_dem_black == 1)$mod1_eligiblility)
black_eligibility[2] <- mean(subset(test, tm1_dem_black == 1)$mod2_eligiblility)
black_eligibility[3] <- mean(subset(test, tm1_dem_black == 1)$mod3_eligiblility)
black_eligibility[4] <- mean(subset(test, tm1_dem_black == 1)$mod4_eligiblility)

# Create a data frame of results
data.frame(algorithm = c("Model 1 - Costs (excl. race) ",
                        "Model 2 - Costs (incl. race) ",
                        "Model 3 - Health (excl. race)",
                        "Model 4 - Health (incl. race)"),
           black_eligibility)

```

7b: Share of Black Patients Eligible

```

##           algorithm black_eligibility
## 1 Model 1 - Costs (excl. race)      0.5119952
## 2 Model 2 - Costs (incl. race)      0.5437825
## 3 Model 3 - Health (excl. race)     0.5921631
## 4 Model 4 - Health (incl. race)     0.6115554

```

```

# Create placeholder matrix
p <- 4
eligible_share_black <- matrix(0, p, 1)

# Calculate mean eligibility
eligible_share_black[1] <- mean(subset(test, mod1_eligiblility == 1)$tm1_dem_black)
eligible_share_black[2] <- mean(subset(test, mod2_eligiblility == 1)$tm1_dem_black)
eligible_share_black[3] <- mean(subset(test, mod3_eligiblility == 1)$tm1_dem_black)
eligible_share_black[4] <- mean(subset(test, mod4_eligiblility == 1)$tm1_dem_black)

# Create a data frame of results
data.frame(algorithm = c("Model 1 - Costs (excl. race) ",
                        "Model 2 - Costs (incl. race) ",
                        "Model 3 - Health (excl. race)",
                        "Model 4 - Health (incl. race)"),
           eligible_share_black)

```

7c: Share of Eligible Patients That Are Black

```

##           algorithm eligible_share_black
## 1 Model 1 - Costs (excl. race)      0.1299472
## 2 Model 2 - Costs (incl. race)      0.1380150
## 3 Model 3 - Health (excl. race)     0.1502943
## 4 Model 4 - Health (incl. race)     0.1552240

```

Question 8: Plotting Key Figures

```
# Plot model 1
ggplot(test, aes(x = risk_cost , y = cost_t, color = race)) +
  stat_binmean(n = 20, geom = "line") +
  stat_binmean(n = 20, geom = "point") +
  geom_vline(xintercept = 55, linetype="dashed") +
  labs(title = "Total Medical Expenditure vs Risk Score [Model 1 - Costs (excl. race)]") +
  theme_minimal() +
  theme(plot.title = element_text(size=20,hjust = 0.5))
```

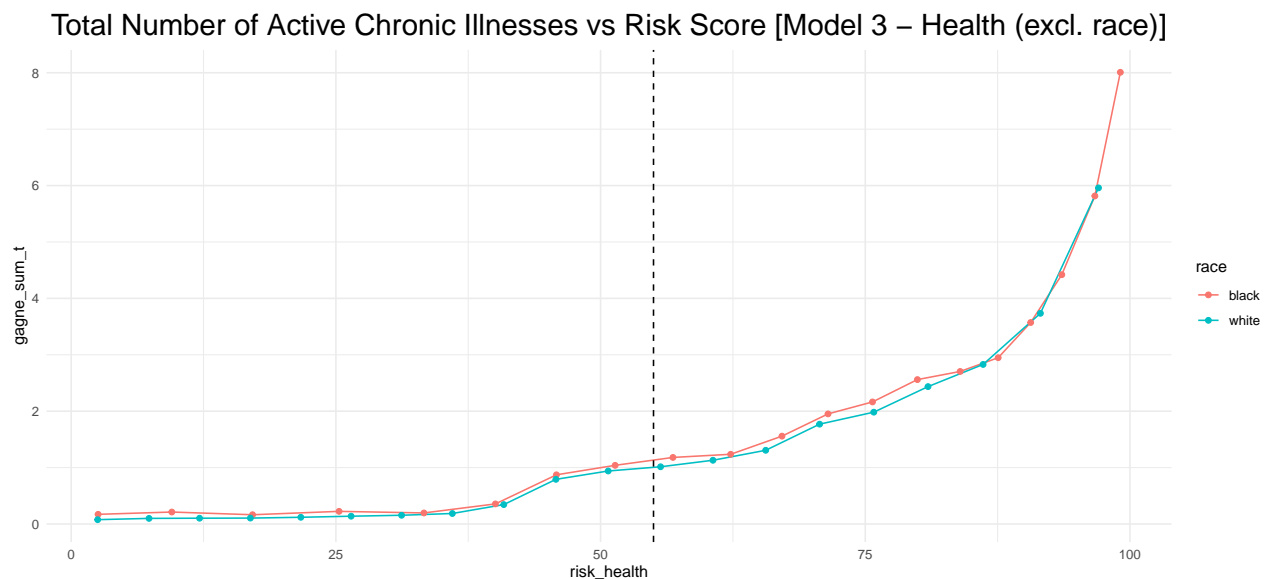


```
# Plot model 2
ggplot(test, aes(x = risk_cost_race , y = cost_t, color = race)) +
  stat_binmean(n = 20, geom = "line") +
  stat_binmean(n = 20, geom = "point") +
  geom_vline(xintercept = 55, linetype="dashed") +
  labs(title = "Total Medical Expenditure vs Risk Score [Model 2 - Costs (incl. race)]") +
  theme_minimal() +
  theme(plot.title = element_text(size=20,hjust = 0.5))
```




Plot model 3

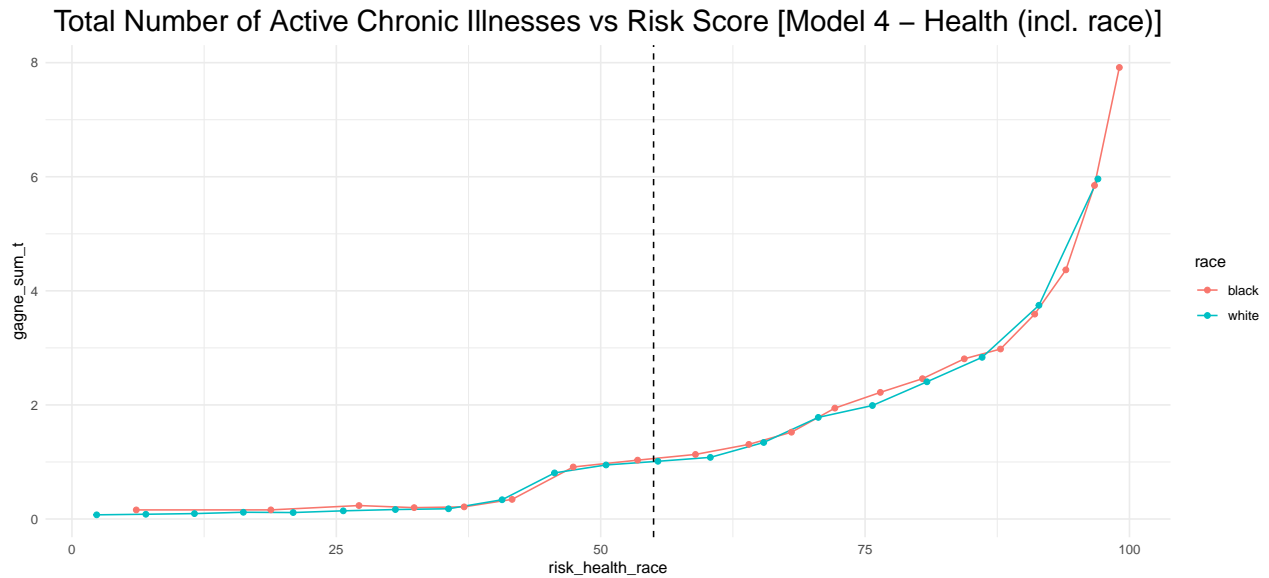
```
ggplot(test, aes(x = risk_health , y = gagne_sum_t, color = race)) +
  stat_binmean(n = 20, geom = "line") +
  stat_binmean(n = 20, geom = "point") +
  geom_vline(xintercept = 55, linetype="dashed") +
  labs(title = "Total Number of Active Chronic Illnesses vs Risk Score [Model 3 – Health (excl. race)]") +
  theme_minimal() +
  theme(plot.title = element_text(size=20,hjust = 0.5))
```



Plot model 4

```
ggplot(test, aes(x = risk_health_race , y = gagne_sum_t, color = race)) +
  stat_binmean(n = 20, geom = "line") +
  stat_binmean(n = 20, geom = "point") +
  geom_vline(xintercept = 55, linetype="dashed") +
```

```
labs(title = "Total Number of Active Chronic Illnesses vs Risk Score [Model 4 – Health (incl. race)]",
theme_minimal() +
theme(plot.title = element_text(size=20,hjust = 0.5))
```



Question 9: Target Parameters and Algorithmic Bias

As professor Obermeyer described in his recorded lecture, an algorithm's bias is determined primarily by the variable that it attempts to predict rather than the vector of predictor variables it takes as inputs to do so (including race and its correlates). Because the predicted variable is almost always a proxy for some other, harder-to-measure real-world outcome, selecting a poor or distorted proxy can introduce significant bias. In the healthcare risk prediction application studied by Obermeyer, the researchers point out that the algorithm was trained to best predict cost of care, which it does well, but fails to account for the fact that cost of care both does not always reflect the need for care and that due to structural societal factors under-accounts for the needs of Black patients in particular. This finding is reflected in the series of scatterplots shown above, where we can see that at the same risk score Black patients tend to have a higher number of total active chronic illnesses than white patients while their total medical expenditure tends to be lower.