# Homework 7

Tobias Boggess

2022-04-02

## Appendix E Problems

**Problem 5: Investigators in the HELP (Health Evaluation and Linkage to Primary Care) study were interested in modeling predictors of being homeless (one or more nights spent on the street or in a shelter in the past six months vs. housed) using baseline data from the clinical trial. Fit and interpret a parsimonious model that would help the investigators identify predictors of homelessness.**

Code:

```
HELPrct <- mutate(HELPrct,
                  homeless01 = case_when(homeless == "housed" ~ 0,
                                         homeless == "homeless" ~ 1))

most_var_HELPrct <-
  glm(
    homeless01 ~ age + cesd + d1 + drugrisk + e2b + i1 + i2 + indtot + mcs + pcs + pss_fr + sexrisk,
    data = HELPrct,
    family = "binomial"
  )

summary(most_var_HELPrct)
```

```
##
## Call:
## glm(formula = homeless01 ~ age + cesd + d1 + drugrisk + e2b +
##     i1 + i2 + indtot + mcs + pcs + pss_fr + sexrisk, family = "binomial",
##     data = HELPrct)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1802  -1.1091   0.6097   0.9137   1.6189
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.227378   2.295251  -0.970   0.3318
## age          0.002978   0.021208   0.140   0.8883
## cesd        -0.007405   0.020198  -0.367   0.7139
## d1           0.010747   0.042860   0.251   0.8020
## drugrisk    -0.003611   0.030996  -0.117   0.9073
```

```
## e2b          0.135400   0.083767   1.616   0.1060
## i1           0.020840   0.020150   1.034   0.3010
## i2           0.002395   0.016154   0.148   0.8821
## indtot       0.075756   0.030570   2.478   0.0132 *
## mcs          0.039673   0.021557   1.840   0.0657 .
## pcs         -0.030173   0.017094  -1.765   0.0775 .
## pss_fr      -0.044783   0.042361  -1.057   0.2904
## sexrisk     -0.011136   0.057217  -0.195   0.8457
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 278.72  on 212  degrees of freedom
## Residual deviance: 245.08  on 200  degrees of freedom
##   (240 observations deleted due to missingness)
## AIC: 271.08
##
## Number of Fisher Scoring iterations: 5
```

```r
sec_most_HELPrct <-
  glm(
    homeless01 ~ age + cesd + d1 + drugrisk + i1 + i2 + indtot + mcs + pcs + pss_fr + sexrisk,
    data = HELPrct,
    family = "binomial"
  )

summary(sec_most_HELPrct)
```

```
##
## Call:
## glm(formula = homeless01 ~ age + cesd + d1 + drugrisk + i1 +
##     i2 + indtot + mcs + pcs + pss_fr + sexrisk, family = "binomial",
##     data = HELPrct)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9907  -1.0405  -0.6291   1.1161   1.9642
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.546992   1.326066  -1.921  0.05477 .
## age          0.015935   0.013924   1.144  0.25247
## cesd        -0.003767   0.011750  -0.321  0.74849
## d1           0.001371   0.016931   0.081  0.93545
## drugrisk     0.017045   0.024569   0.694  0.48783
## i1           0.031015   0.010757   2.883  0.00394 **
## i2          -0.007138   0.007097  -1.006  0.31450
## indtot       0.052587   0.017300   3.040  0.00237 **
## mcs          0.004855   0.011283   0.430  0.66697
## pcs         -0.004756   0.010344  -0.460  0.64566
## pss_fr      -0.075931   0.026304  -2.887  0.00389 **
## sexrisk      0.050037   0.037150   1.347  0.17801
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 624.05  on 451  degrees of freedom
## Residual deviance: 569.03  on 440  degrees of freedom
##   (1 observation deleted due to missingness)
## AIC: 593.03
##
## Number of Fisher Scoring iterations: 4
```

```
thrd_most_HELPrct <-
  glm(homeless01 ~ i1 + indtot + pss_fr,
      data = HELPrct,
      family = "binomial")

summary(thrd_most_HELPrct)
```

```
##
## Call:
## glm(formula = homeless01 ~ i1 + indtot + pss_fr, family = "binomial",
##     data = HELPrct)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9384  -1.0406  -0.6929   1.1410   1.8927
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.866940   0.619284  -3.015  0.00257 **
## i1           0.023155   0.005778   4.007 6.14e-05 ***
## indtot       0.049282   0.015773   3.124  0.00178 **
## pss_fr      -0.070610   0.025569  -2.762  0.00575 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
## Residual deviance: 575.99  on 449  degrees of freedom
## AIC: 583.99
##
## Number of Fisher Scoring iterations: 4
```

```
fth_most_HELPrct <-
  glm(homeless01 ~ i1 + pss_fr, data = HELPrct, family = "binomial")

summary(fth_most_HELPrct)
```

```
##
## Call:
## glm(formula = homeless01 ~ i1 + pss_fr, family = "binomial",
##     data = HELPrct)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0043  -1.0532  -0.7962   1.1751   1.7046
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.058573   0.214945  -0.273  0.78524
## i1           0.026414   0.005733   4.608 4.07e-06 ***
## pss_fr      -0.084327   0.025045  -3.367  0.00076 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
## Residual deviance: 586.45  on 450  degrees of freedom
## AIC: 592.45
##
## Number of Fisher Scoring iterations: 4
```

```
fifth_most_HELPrct <-
  glm(homeless01 ~ i1 + indtot, data = HELPrct, family = "binomial")

summary(fifth_most_HELPrct)
```

```
##
## Call:
## glm(formula = homeless01 ~ i1 + indtot, family = "binomial",
##     data = HELPrct)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7925  -1.0533  -0.6767   1.1574   1.9634
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.578277   0.563927  -4.572 4.83e-06 ***
## i1           0.023465   0.005739   4.089 4.34e-05 ***
## indtot       0.055891   0.015452   3.617 0.000298 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
## Residual deviance: 583.75  on 450  degrees of freedom
## AIC: 589.75
##
## Number of Fisher Scoring iterations: 4
```

```
six_most_HELPrct <-
  glm(homeless01 ~ pss_fr + indtot, data = HELPrct, family = "binomial")
```

```
summary(six_most_HELPrct)
```

```
##
## Call:
## glm(formula = homeless01 ~ pss_fr + indtot, family = "binomial",
##     data = HELPrct)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5360  -1.0921  -0.7134   1.1297   2.1035
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.88379    0.61990  -3.039  0.00237 **
## pss_fr      -0.07209    0.02507  -2.875  0.00404 **
## indtot       0.06128    0.01566   3.914 9.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
## Residual deviance: 594.47  on 450  degrees of freedom
## AIC: 600.47
##
## Number of Fisher Scoring iterations: 4
```

The best of the models seems to be with the explanatory variables i1 (average number of drinks) and indtot (inventory of drug use consequences total score). OVerall, the two variables make sense of why someone would be homeless because the i1 variable takes the average amount of drinks that are consumed by a person in a day. The other variable, indtot, takes into account effects of continued drug usage from an individual questionare that is scored based on several questions applicants answer. While I have not included another variable, pss_fr which is social support from friends, may help in predicting homelessness.

## Chapter 10 Problems

**Problem 3: Do the following using the HELP data set.**

**a) Generate a confusion matrix for the null model and interpret the result.**
Code:

```
my.logreg <- glm(homeless01 ~ 1, data = HELPrct, family = "binomial")

logreg.prob <-
  predict(my.logreg, newdata = HELPrct, type = "response")

HELPrct <-
  mutate(HELPrct,
         predType = case_when(homeless == "housed" ~ 0,
                              homeless == "homeless" ~ 1))
```

```r
conf_matrix <- data.frame(Homeless = 0, Housed = 0)
conf_matrix <- rbind(conf_matrix, table(HELPrct$homeless))
rownames(conf_matrix) <- c("Predicted Homeless", "Predicted Housed")
conf_matrix
```

```
##                    Homeless Housed
## Predicted Homeless        0      0
## Predicted Housed        209    244
```

```r
# conf_mat(HELPrct, truth = homeless01, estimate = predType)
```

**b) Fit and interpret logistic regression model for the probability of being homeless as a function of age.**
Code:

```r
logreg.age <-
  glm(homeless01 ~ age, data = HELPrct, family = "binomial")

summary(logreg.age)
```

```
##
## Call:
## glm(formula = homeless01 ~ age, family = "binomial", data = HELPrct)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.328  -1.106  -1.024   1.231   1.409
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.95724    0.45094  -2.123   0.0338 *
## age          0.02248    0.01234   1.822   0.0685 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 625.28  on 452  degrees of freedom
## Residual deviance: 621.93  on 451  degrees of freedom
## AIC: 625.93
##
## Number of Fisher Scoring iterations: 4
```

For every additional age in years means there is a slightly better chance of being homeless given the model. By this I mean the equation for the probability of being homeless is: P(homeless) = -0.95724 + 0.02248 * *Age*.

**c) What is the predicted probability of being homeless for a 20 year old? For a 40 year old?**
Code:

```r
newHELPrct <- data.frame(age = c(20, 40))

preds <-
  predict(logreg.age, newdata = newHELPrct, type = "response")
preds
```

```
##         1         2
## 0.3757450 0.4854891
```

The predicted probability of being homeless for the 20 year old is approximately 37.57% and the probability of being homeless for a 40 year old is 48.55%.

**d) Generate a confusion matrix for the second model and interpret the result.**
Code:

```r
probs.age <-
  predict(logreg.age, newdata = HELPrct, type = "response")

preds.age <- case_when(probs.age < 0.5 ~ "housed",
                       probs.age >= 0.5 ~ "homeless")

HELPrct <- mutate(HELPrct, predType = preds.age)

conf_mat(data = HELPrct,
         truth = homeless,
         estimate = predType)
```

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...):
## `estimate` was converted to a factor
```

```
##             Truth
## Prediction homeless housed
##   homeless       48     35
##   housed        161    209
```

Based on the confusion matrix, the number of housed people predicted correctly is 209 while 35 were incorrectly labeled as homeless. The amount of homeless correctly predicted is 48 while 161 homeless were labeled as housed. In other words, the accuracy of the predictions isn't very good for the homeless but decent for the housed population.

## Chapter 11 Problems

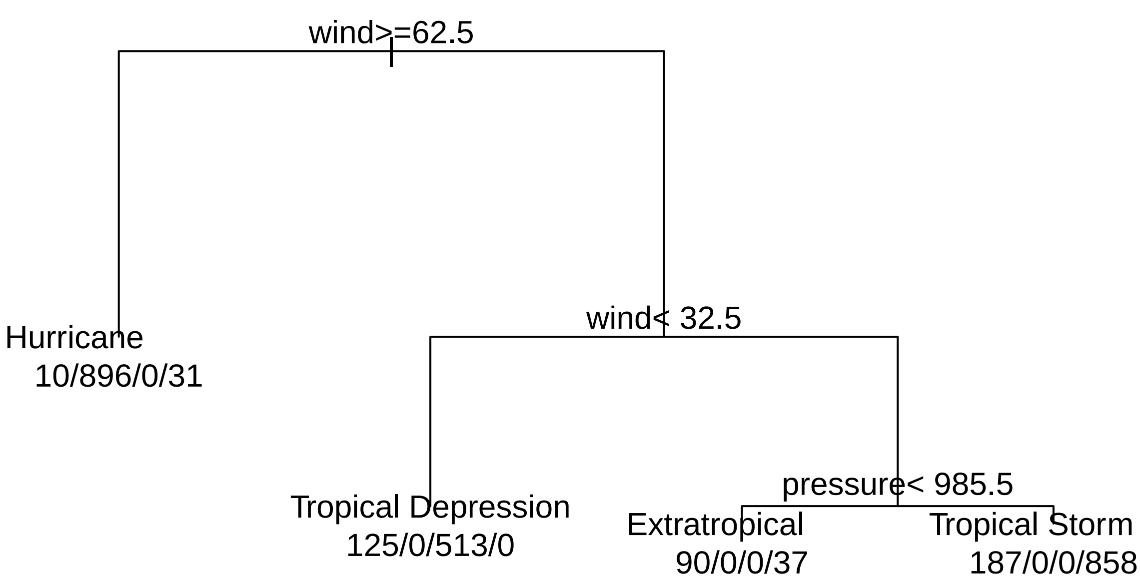
**Problem 4: Build a classifier for the type of each storm as a function of its wind speed and pressure. Why would a decision tree make a particularly good classifier for these data? Visualize your classifier in the data space.**

Code:

```r
storms.tree <- rpart(type ~ wind + pressure,
                     data = storms,
                     control = rpart.control(minsplit = 12))

storms.tree
```

```
## n= 2747
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 2747 1821 Tropical Storm (0.14998180 0.32617401 0.18674918 0.33709501)
##    2) wind>=62.5 937   41 Hurricane (0.01067236 0.95624333 0.00000000 0.03308431) *
##    3) wind< 62.5 1810  915 Tropical Storm (0.22209945 0.00000000 0.28342541 0.49447514)
##      6) wind< 32.5 638  125 Tropical Depression (0.19592476 0.00000000 0.80407524 0.00000000) *
##      7) wind>=32.5 1172  277 Tropical Storm (0.23634812 0.00000000 0.00000000 0.76365188)
##       14) pressure< 985.5 127   37 Extratropical (0.70866142 0.00000000 0.00000000 0.29133858) *
##       15) pressure>=985.5 1045  187 Tropical Storm (0.17894737 0.00000000 0.00000000 0.82105263) *
```

```r
par(xpd = TRUE)

plot(storms.tree, compress = TRUE)
text(storms.tree, use.n = TRUE)
```



```r
par(xpd = FALSE)

storms.pred <- predict(storms.tree, type = "class")

storms <- mutate(storms, predType = storms.pred)

accuracy(data = storms,
         truth = as.factor(type),
         estimate = as.factor(predType))
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.858
```

```r
conf_mat(data = storms,
         truth = type,
         estimate = predType)
```

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...):
## `truth` was converted to a factor
```

```
##                        Truth
## Prediction              Extratropical Hurricane Tropical Depression
##   Extratropical                    90         0                   0
##   Hurricane                        10       896                   0
##   Tropical Depression             125         0                 513
##   Tropical Storm                  187         0                   0
##                        Truth
## Prediction              Tropical Storm
##   Extratropical                     37
##   Hurricane                         31
##   Tropical Depression                0
##   Tropical Storm                   858
```

```r
storms.forest <- randomForest(
  as.factor(type) ~ wind + pressure,
  data = storms,
  ntree = 500,
  mtry = 2
)

storms.forest
```

```
##
## Call:
##  randomForest(formula = as.factor(type) ~ wind + pressure, data = storms,      ntree = 500, mtry = 2
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
```

```
##         OOB estimate of  error rate: 12.63%
## Confusion matrix:
##                     Extratropical Hurricane Tropical Depression Tropical Storm
## Extratropical                 189         8                  80            135
## Hurricane                       3       886                   0              7
## Tropical Depression            20         0                 493              0
## Tropical Storm                 65        29                   0            832
##                     class.error
## Extratropical        0.54126214
## Hurricane            0.01116071
## Tropical Depression  0.03898635
## Tropical Storm       0.10151188
```

```
storms.forest.pred <- predict(storms.forest, type = "class")
storms <- mutate(storms, predType = storms.forest.pred)
accuracy(data = storms,
         truth = as.factor(type),
         estimate = as.factor(predType))
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.874
```

```
conf_mat(data = storms,
         truth = type,
         estimate = predType)
```

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...):
## `truth` was converted to a factor
```

```
##                      Truth
## Prediction            Extratropical Hurricane Tropical Depression
##   Extratropical                 189         3                  20
##   Hurricane                       8       886                   0
##   Tropical Depression            80         0                 493
##   Tropical Storm                135         7                   0
##                      Truth
## Prediction            Tropical Storm
##   Extratropical                   65
##   Hurricane                       29
##   Tropical Depression              0
##   Tropical Storm                 832
```

A decision tree is the best for this type of problem because it has less misclassification for each type of storm. The decision tree is also less complex than the randomForest. The best part is the accuracy of the predictions is around 85.8% which is a pretty good prediction accuracy probability.

**Problem 6 (only a and c; decision tree, random forest, and k-NN): Do the following.**

**a) For each of the following models:** • Build a classifier for SleepTrouble • Report its effectiveness on the NHANES training data • Make an appropriate visualization of the model • Interpret the results. What

have you learned about people's sleeping habits?
Decision tree code:

```
nhanes1 <-
  select(
    .data = NHANES,
    SleepTrouble,
    Age,
    Poverty,
    BMI,
    BPSysAve,
    BPDiaAve,
    TotChol,
    Weight,
    Height,
    Pulse,
    HomeRooms,
    SleepHrsNight
  )

nhanes1 <- na.omit(nhanes1)
```

```
nhanes1.tree <-
  rpart(
    SleepTrouble ~ Age + Poverty + BMI + Weight + Height,
    data = nhanes1,
    control = rpart.control(cp = 0.15)
  )
summary(nhanes1.tree)
```

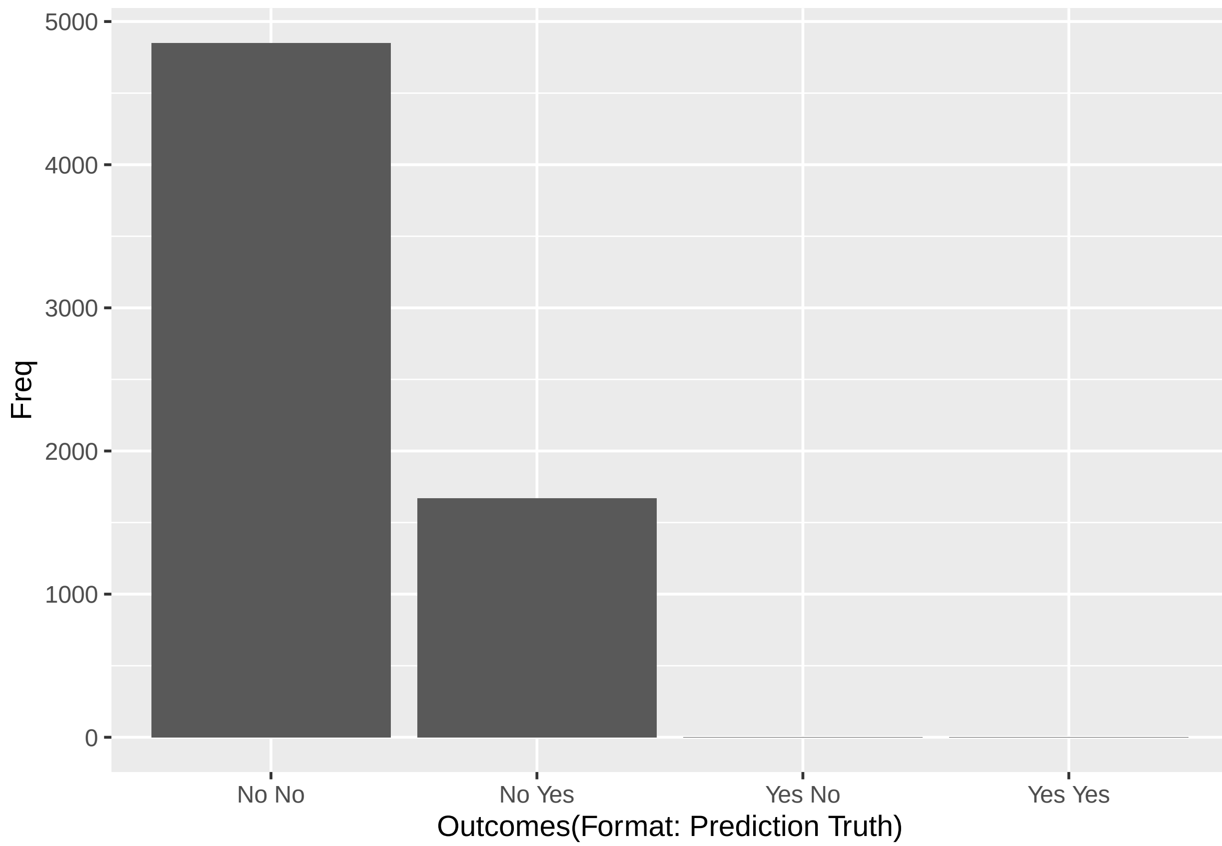
```
## Call:
## rpart(formula = SleepTrouble ~ Age + Poverty + BMI + Weight +
##     Height, data = nhanes1, control = rpart.control(cp = 0.15))
##   n= 6520
##
##   CP nsplit rel error xerror xstd
## 1  0      0         1      0    0
##
## Node number 1: 6520 observations
##   predicted class=No  expected loss=0.2559816  P(node) =1
##     class counts:  4851  1669
##    probabilities: 0.744 0.256
```

```
nhanes1.pred <- predict(nhanes1.tree, type = "class")
nhanes1.df <- mutate(.data = nhanes1, predType = nhanes1.pred)

tree.conf <-
  conf_mat(data = nhanes1.df,
           truth = SleepTrouble,
           estimate = predType)
tree.conf
```

```
##             Truth
```

```
## Prediction   No  Yes
##        No  4851 1669
##        Yes    0    0
```

```
tree.conf <- data.frame(tree.conf$table)
tree.conf <- unite(data = tree.conf,
                    col = "Combo",
                    c(Prediction, Truth),
                    sep = " ")

ggplot(data = tree.conf) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcomes(Format: Prediction Truth)")
```




Random Forest Code:

```
nhanes1.forest <-
  randomForest(
    SleepTrouble ~ Age + BMI + Poverty + Pulse + Height + Weight + TotChol,
    data = nhanes1,
    ntree = 1000,
    mtry = 3
  )

nhanes1.forest
```

```
##
## Call:
##  randomForest(formula = SleepTrouble ~ Age + BMI + Poverty + Pulse +      Height + Weight + TotChol,
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 10.9%
## Confusion matrix:
##       No  Yes class.error
## No  4744  107  0.02205731
## Yes  604 1065  0.36189335
```

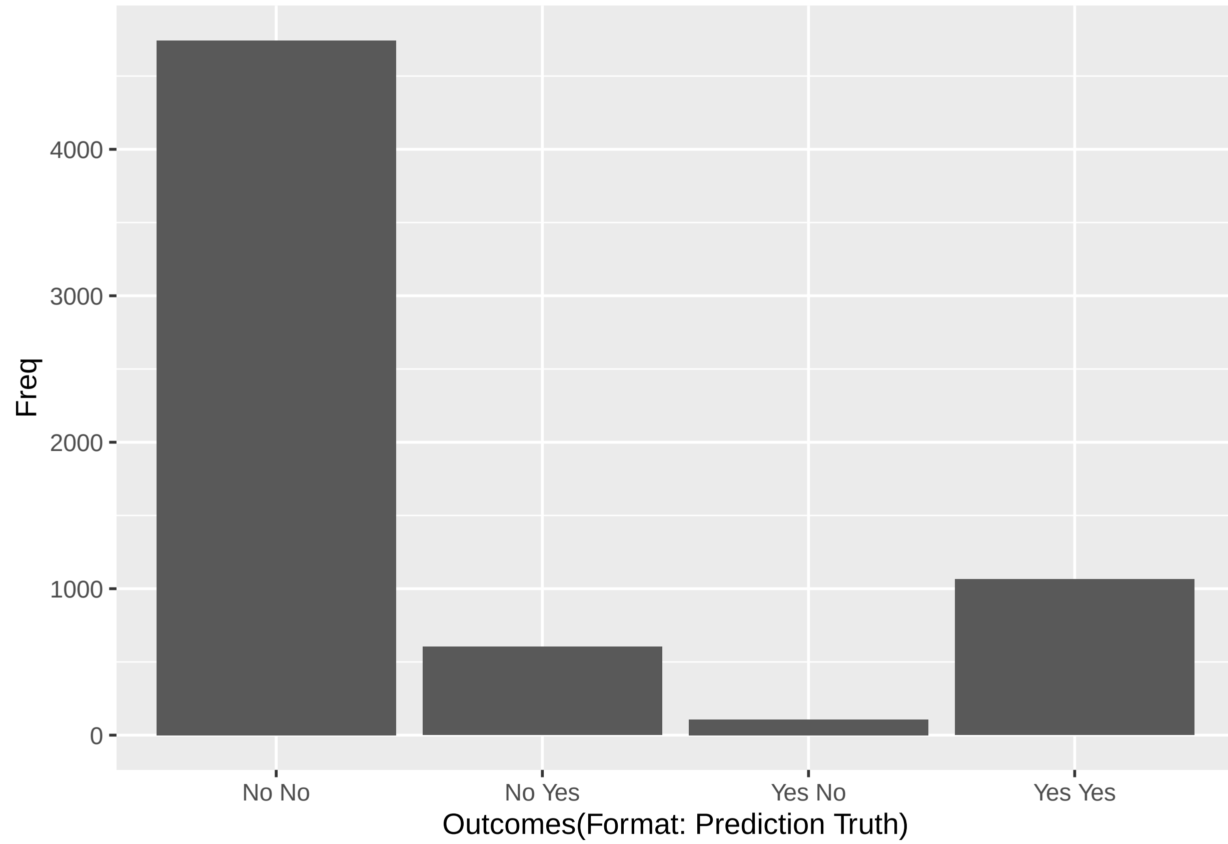
```
nhanes.pred.forest <- predict(nhanes1.forest, type = "class")
nhanes2 <- mutate(nhanes1, predType = nhanes.pred.forest)

accuracy(
  data = nhanes2,
  truth = as.factor(SleepTrouble),
  estimate = as.factor(predType)
)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.891
```

```
for.conf.mat <-
  conf_mat(data = nhanes2,
           truth = SleepTrouble,
           estimate = predType)
for.conf.mat
```

```
##           Truth
## Prediction   No  Yes
##        No  4744  604
##        Yes  107 1065
```

```
for.conf.mat <- data.frame(for.conf.mat$table)
for.conf.mat <- unite(data = for.conf.mat,
        col = "Combo",
        c(Prediction, Truth),
        sep = " ")

ggplot(data = for.conf.mat) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcomes(Format: Prediction Truth)")
```

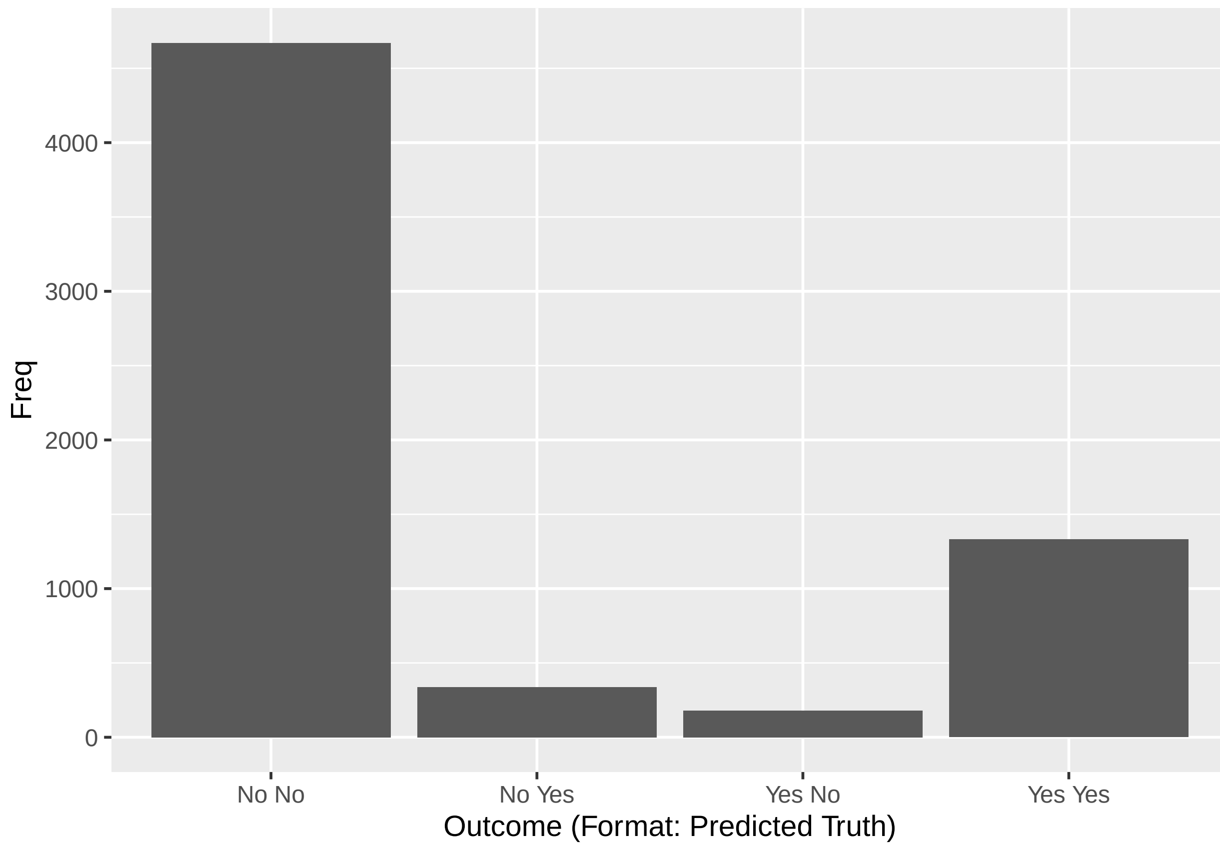



K-NN Code:

```
nhanes1.knn <-
  kknn(
    SleepTrouble ~ Age + BMI + Poverty + Pulse + Height + Weight + TotChol,
    train = nhanes1,
    test = nhanes1,
    k = 7
  )

knn.preds <- fitted(nhanes1.knn)
nhanes3 <- mutate(nhanes1, predType = knn.preds)

accuracy(
  data = nhanes3,
  truth = as.factor(SleepTrouble),
  estimate = as.factor(predType)
)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.921
```

```
knn.conf_mat <-
  conf_mat(data = nhanes3,
           truth = SleepTrouble,
           estimate = predType)
knn.conf_mat <- data.frame(knn.conf_mat$table)
knn.conf_mat <- unite(knn.conf_mat,
        col = "Combo",
        c(Prediction, Truth),
        sep = " ")

ggplot(data = knn.conf_mat) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcome (Format: Predicted Truth)")
```




**c) Repeat either of the previous exercises, but this time first separate the NHANES data set uniformly at random into 75% training and 25% testing sets. Compare the effectiveness of each model on training vs. testing data.**

Decision Tree:

```
nhanes.data = sort(sample(nrow(nhanes1), nrow(nhanes1)*.75))
nhanes1.train <- nhanes1[nhanes.data,]
nhanes1.test <- nhanes1[-nhanes.data,]

nhanes2.tree <-
```

```r
  rpart(
    SleepTrouble ~ Age + BMI + Poverty + Pulse + Height + Weight + TotChol,
    data = nhanes1.train,
    control = rpart.control(cp = 0.15)
  )

summary(nhanes2.tree)
```

```
## Call:
## rpart(formula = SleepTrouble ~ Age + BMI + Poverty + Pulse +
##     Height + Weight + TotChol, data = nhanes1.train, control = rpart.control(cp = 0.15))
##   n= 4890
##
##   CP nsplit rel error xerror xstd
## 1  0      0         1      0    0
##
## Node number 1: 4890 observations
##   predicted class=No  expected loss=0.2537832  P(node) =1
##     class counts:  3649  1241
##    probabilities: 0.746 0.254
```

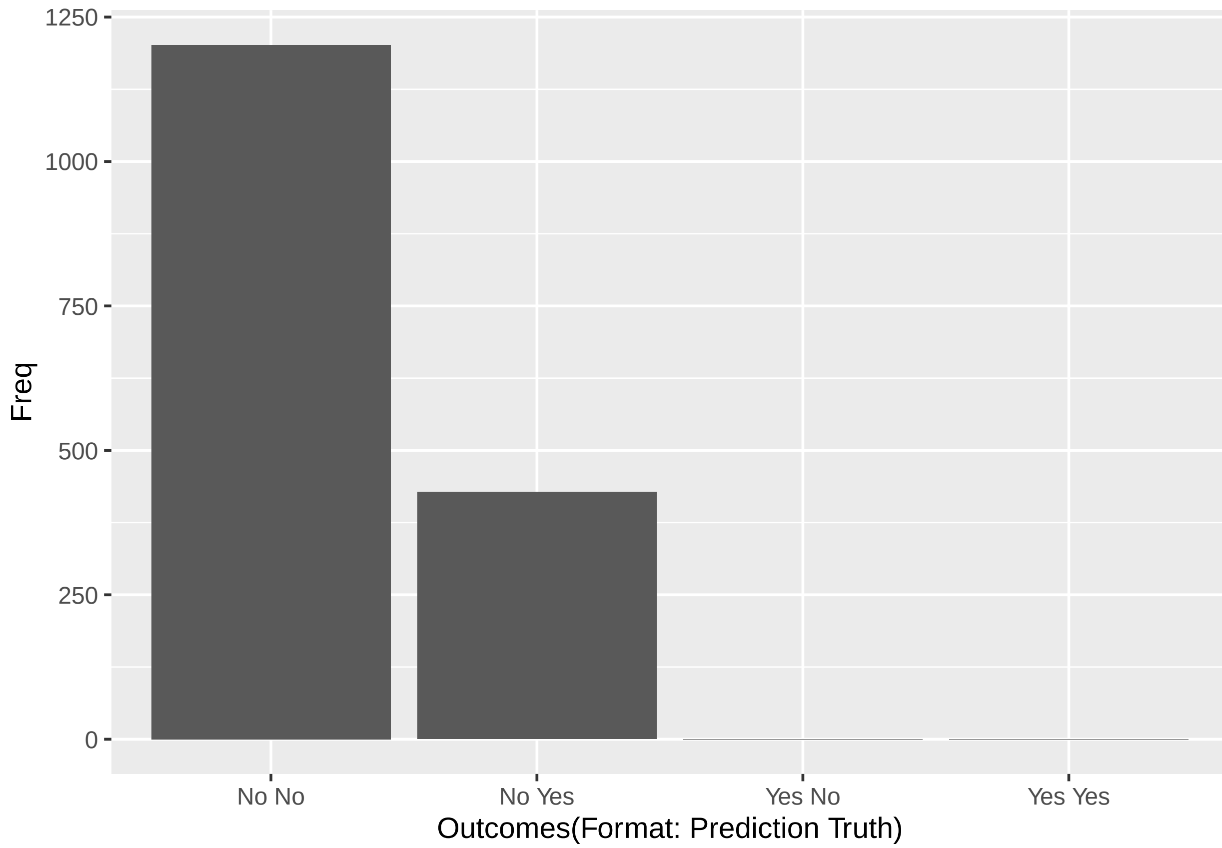
```r
# nhanes2.tree
tree.preds <- predict(nhanes2.tree, newdata = nhanes1.test, type = "class")
nhanes2.test <- mutate(nhanes1.test, predType = tree.preds)
accuracy(data = nhanes2.test, truth = SleepTrouble, estimate = predType)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.737
```

```r
tree.conf1 <-
  conf_mat(data = nhanes2.test,
           truth = SleepTrouble,
           estimate = predType)
tree.conf1
```

```
##           Truth
## Prediction   No  Yes
##        No  1202  428
##        Yes    0    0
```

```r
tree.conf1 <- data.frame(tree.conf1$table)
tree.conf1 <- unite(data = tree.conf1,
                    col = "Combo",
                    c(Prediction, Truth),
                    sep = " ")

ggplot(data = tree.conf1) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcomes(Format: Prediction Truth)")
```




Random Forest:

```
nhanes2.forest <-
  randomForest(
    SleepTrouble ~ Age + BMI + Poverty + Pulse + Height + Weight + TotChol,
    data = nhanes1.train,
    ntree = 1000,
    mtry = 3
  )

nhanes2.forest
```

```
##
## Call:
##  randomForest(formula = SleepTrouble ~ Age + BMI + Poverty + Pulse +      Height + Weight + TotChol,
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 12.66%
## Confusion matrix:
##       No Yes class.error
## No  3561  88   0.0241162
## Yes  531 710   0.4278807
```

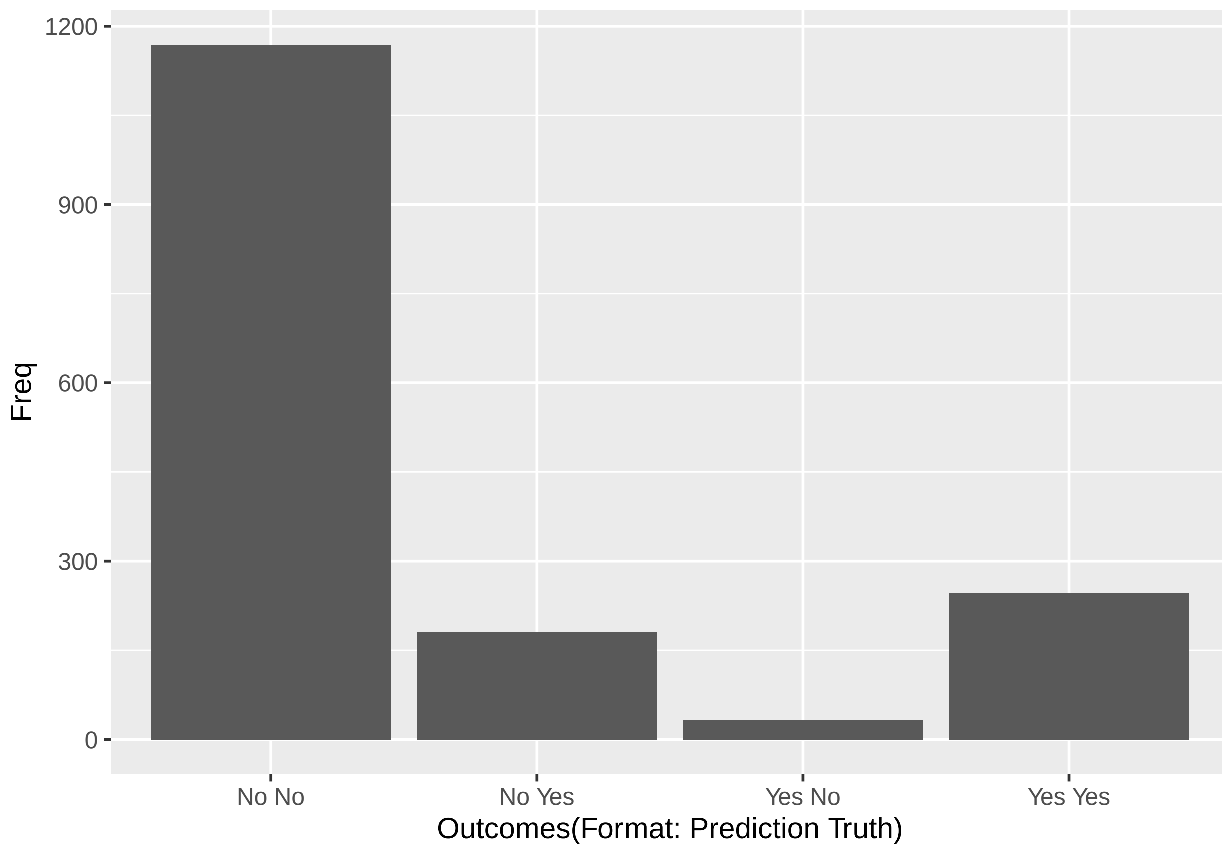
```r
forest.preds <-
  predict(nhanes2.forest, newdata = nhanes1.test, type = "class")
nhanes2.test <- mutate(nhanes1.test, predType = forest.preds)
accuracy(data = nhanes2.test,
         truth = SleepTrouble,
         estimate = predType)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.869
```

```r
for.conf.mat1 <-
  conf_mat(data = nhanes2.test,
           truth = SleepTrouble,
           estimate = predType)
for.conf.mat1
```

```
##           Truth
## Prediction   No  Yes
##        No  1169  181
##        Yes   33  247
```

```r
for.conf.mat1 <- data.frame(for.conf.mat1$table)
for.conf.mat1 <- unite(data = for.conf.mat1,
        col = "Combo",
        c(Prediction, Truth),
        sep = " ")

ggplot(data = for.conf.mat1) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcomes(Format: Prediction Truth)")
```

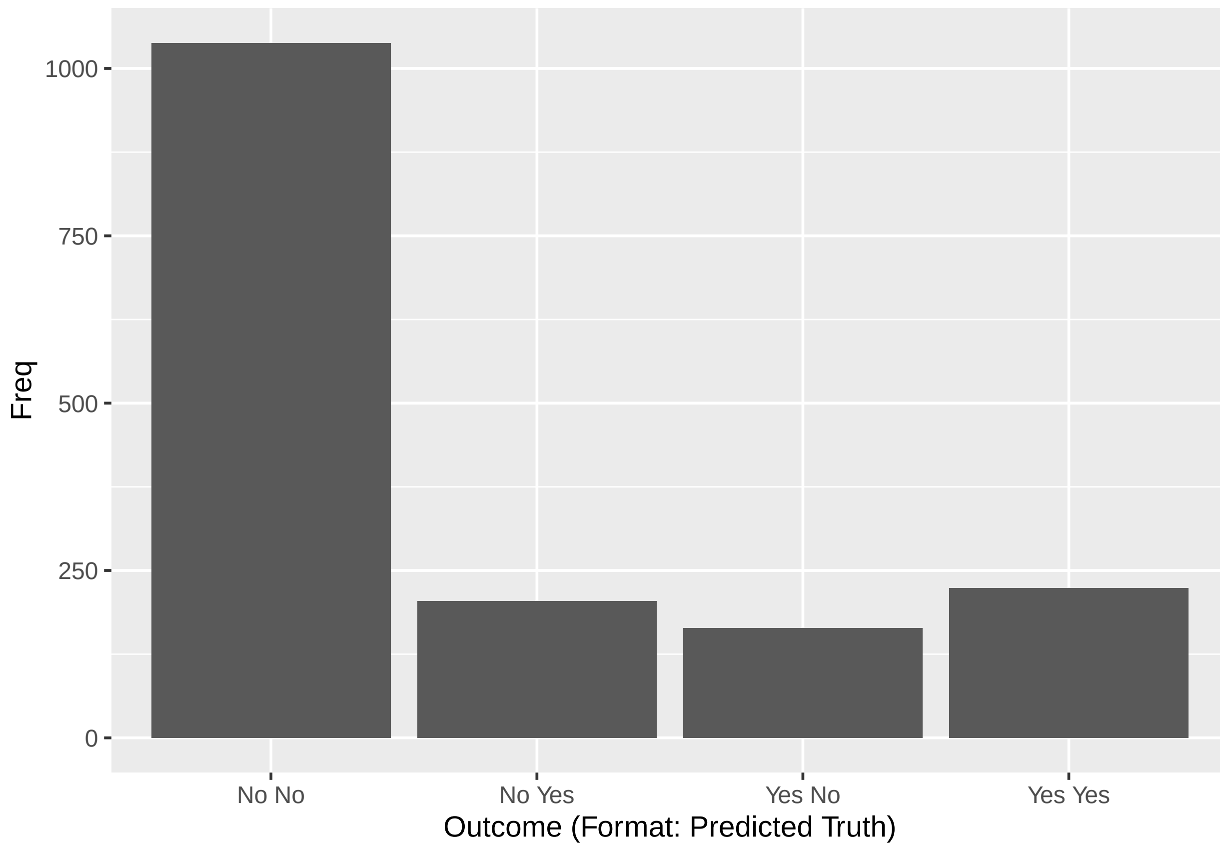



K-Nearest Neighbor:

```
nhanes3.knn <-
  kknn(
    SleepTrouble ~ Age + BMI + Poverty + Pulse + Height + Weight + TotChol,
    train = nhanes1.train,
    test = nhanes1.test,
    k = 7
  )

knn.preds1 <- fitted(nhanes3.knn)
nhanes3.test <- mutate(nhanes1.test, predType = knn.preds1)

accuracy(
  data = nhanes3.test,
  truth = as.factor(SleepTrouble),
  estimate = as.factor(predType)
)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.774
```

```
knn.conf_mat1 <-
  conf_mat(data = nhanes3.test,
           truth = SleepTrouble,
           estimate = predType)
knn.conf_mat1 <- data.frame(knn.conf_mat1$table)
knn.conf_mat1 <- unite(knn.conf_mat1,
        col = "Combo",
        c(Prediction, Truth),
        sep = " ")

ggplot(data = knn.conf_mat1) +
  geom_col(mapping = aes(x = Combo, y = Freq)) +
  xlab("Outcome (Format: Predicted Truth)")
```




The models using a training set of the data seem to be a better representation of the data and gives a better estimate of the different models.