

# Question 2

## Question 2A

Restrict to stops where the suspected.crime is 'cpw', then train a logistic regression model on all of 2008, predicting whether or not a weapon is found. Use the following features as predictors, standardizing real-valued attributes.

```
#Filter to only to crime = cpw
sqf.data <- sqf_08_16 %>% filter(suspected.crime=='cpw')

#Convert variable types as necessary
sqf.data <- sqf.data %>% mutate(suspect.race = as.factor(suspect.race),
                               suspect.build = as.factor(suspect.build),
                               suspect.sex = as.factor(suspect.sex),
                               location.housing = as.factor(location.housing),
                               day = as.factor(day),
                               month = as.factor(month),
                               time.period = as.factor(time.period),
                               precinct = as.factor(precinct))

# Calculting these for a later problem.
age <- (30-mean(sqf.data$suspect.age, na.rm = TRUE))/sd(sqf.data$suspect.age, na.rm=TRUE)

weight <- (165-mean(sqf.data$suspect.weight, na.rm = TRUE))/sd(sqf.data$suspect.weight, na.rm = TRUE)

height <- (6-mean(sqf.data$suspect.height, na.rm = TRUE))/sd(sqf.data$suspect.height, na.rm = TRUE)

op <- (10-mean(sqf.data$observation.period, na.rm = TRUE))/sd(sqf.data$observation.period, na.rm = TRUE)

#Standardize real-valued variables
sqf.data <- sqf.data %>%
  mutate(suspect.height = standardize(suspect.height),
         suspect.weight = standardize(suspect.weight),
         suspect.age = standardize(suspect.age),
         observation.period = standardize(observation.period))

#Filter to only year = 2008. This is our training set.
sqf.data.2008 <- sqf.data %>% filter(year == 2008)

#Filter to only year = 2009. This is our test set.
sqf.data.2009 <- sqf.data %>% filter(year == 2009)

#Logistic Model
sqf2008_model <- glm(formula = found.weapon ~ 1 + precinct + location.housing + additional.report +
                     additional.investigation + additional.proximity + additional.evasive +
                     additional.associating + additional.direction + additional.highcrime + additional.
                     additional.sights + additional.other + stopped.bc.object + stopped.bc.desc +
                     stopped.bc.casing + stopped.bc.lookout + stopped.bc.clothing + stopped.bc.drugs +
                     stopped.bc.furtive + stopped.bc.violent + stopped.bc.bulge + stopped.bc.other +
                     suspect.age + suspect.sex + suspect.build + suspect.height + suspect.weight +
                     inside + radio.run + observation.period + day + month + time.period,
```

```
family = "binomial",data = sqf.data.2008)
```

## Question 2AI

What are the ten largest and ten smallest coefficients? Pick one of these coefficients, and give a precise statement as to how that coefficient can be interpreted.

```
# Smallest
(smallest.coefficients<- sqf2008_model$coefficients %>%
  data.frame(VarName = names(.),Coefficients = .) %>%
  arrange(Coefficients) %>%
  slice(1:10))
```

```
##      VarName Coefficients
## 1 (Intercept)  -4.2393018
## 2 precinct44   -1.2451549
## 3 precinct50   -1.1824933
## 4 precinct24   -1.0873367
## 5 precinct101  -1.0224301
## 6 precinct52   -0.9977487
## 7 precinct41   -0.8458078
## 8 precinct75   -0.8036449
## 9 precinct23   -0.7752557
## 10 precinct115 -0.7724176
```

```
# Largest
(largest.coefficients<-
sqf2008_model$coefficients %>%
  data.frame(VarName = names(.),Coefficients = .) %>%
  arrange(desc(Coefficients)) %>%
  slice(1:10))
```

```
##      VarName Coefficients
## 1 location.housingtransit 2.5868688
## 2 stopped.bc.objectTRUE 2.4671972
## 3 precinct111 1.4026402
## 4 precinct25 0.9789911
## 5 location.housingneither 0.8276274
## 6 precinct9 0.7955521
## 7 precinct112 0.7852394
## 8 precinct17 0.7636204
## 9 stopped.bc.otherTRUE 0.7322510
## 10 precinct13 0.6490235
```

## Answer 2AI

The log odds of a stopped person carrying a weapon is increased by 2.59 if they are stopped in transit compared to if they were stopped at home, holding everything else constant.

## Question 2AII

Suppose my (imaginary) friend, a 30 year old, six-foot tall, 165 lb man of medium build was stopped in the West 4th subway station on 10/4/2018 at 8pm (no weapon was found). Upon reviewing the UF-250 form

filled out for his stop, you notice that he was suspected of criminal possession of a weapon, and was stopped because he had a suspicious bulge in his coat, and he was near a part of the station known for having a high incidence of weapon offenses. He was observed for 10 minutes before the stop was made, and the stop was not the result of a radio call. If your model were used to predict the ex-ante probability that my friend were carrying a weapon, what would this probability be? What if my friend were a woman, everything else being equal?

## Answer 2AII

```
#create data frame from model coefficients

results1 <- as.data.frame(coef(sqf2008_model))
results <- transpose(results1)
colnames(results) <- rownames(results1)
names(results)[1] <- "Intercept"

#plug values into logit
logit = results$Intercept + results$location.housingtransit + results$additional.highcrimeTRUE + results$

#calculate probability for male
odds <- exp(logit)

paste('According to the model, the probability that he would be found with a weapon is',
      (odds / (1 + odds)))

## [1] "According to the model, the probability that he would be found with a weapon is 0.1901658511866"

#plug values into logit (adding coefficient for female)
logitf = logit - results$suspect.sexmale

odds1 <- exp(logitf)

paste('According to the model, the probability that SHE would be found with a weapon is',
      (odds1 / (1 + odds1)))

## [1] "According to the model, the probability that SHE would be found with a weapon is 0.164882003553"
```

## Question 2AIII

Compute the AUC of this model on all data from 2009, using the ROCR package (as in lecture 6).

## Answer 2AIII

```
sqf.data.2009$predicted.probability <- predict(sqf2008_model, newdata = sqf.data.2009, type='response')

#AUC
test.pred <- prediction(sqf.data.2009$predicted.probability, sqf.data.2009$found.weapon)
test.perf <- performance(test.pred, "auc")
cat('The AUC score is ', 100*test.perf@y.values[[1]], "\n")

## The AUC score is 81.11205
```

## Question 2AIV

The AUC can be interpreted as the probability that a randomly chosen true instance will be ranked higher than a randomly chosen false instance. Check that this interpretation holds by sampling (with replacement) 10,000 random pairs of true (weapon is found) and false (weapon is not found) examples from 2009, and computing the proportion of pairs where your model predicts that the true example is more likely to find a weapon than the false example. Confirm that your answer is approximately equal to the answer computed in part III)

## Answer 2AIV

```
set.seed(2009)
not.na.idx <- which(!is.na(sqf.data.2009$predicted.probability))
not.na.rows.2009 <- sqf.data.2009[not.na.idx,]
found.weapon.idx <- which(not.na.rows.2009$found.weapon == TRUE)
notfound.weapon.idx <- which(not.na.rows.2009$found.weapon == FALSE)

True.2009.sample <- sample(found.weapon.idx, 10000, replace = TRUE)
False.2009.sample <- sample(notfound.weapon.idx, 10000, replace = TRUE)

ProbCompare <- not.na.rows.2009$predicted.probability[True.2009.sample] > not.na.rows.2009$predicted.pr
mean(ProbCompare)
```

```
## [1] 0.8122
```

The answer we get here is ~81.22% versus ~81.11205% in part 2AIII, making the numbers comparable and leading us to believe that the interpretation holds.

## Question 2B

Using the same model from part A, make a plot where the x-axis shows each year from 2009-2016, and the y-axis shows the model AUC (computed using the ROCR package) when that year is used as the test set. Explain in a few sentences what you observe, why you think this might be happening, and what one might do about it.

## Answer 2B

```
#Create a dataframe with the two axis
model2008_AUCs <- data.frame(year = 2009:2016, AUC = rep(NA,8))

#Keep only the precinct levels consistent from 2008
Precincts.2008 <- unique(sqf.data.2008$precinct)

#Calculate AUC for every year
for(i in model2008_AUCs$year){
  temp.df <- sqf.data %>% filter(year == i, precinct %in% Precincts.2008)

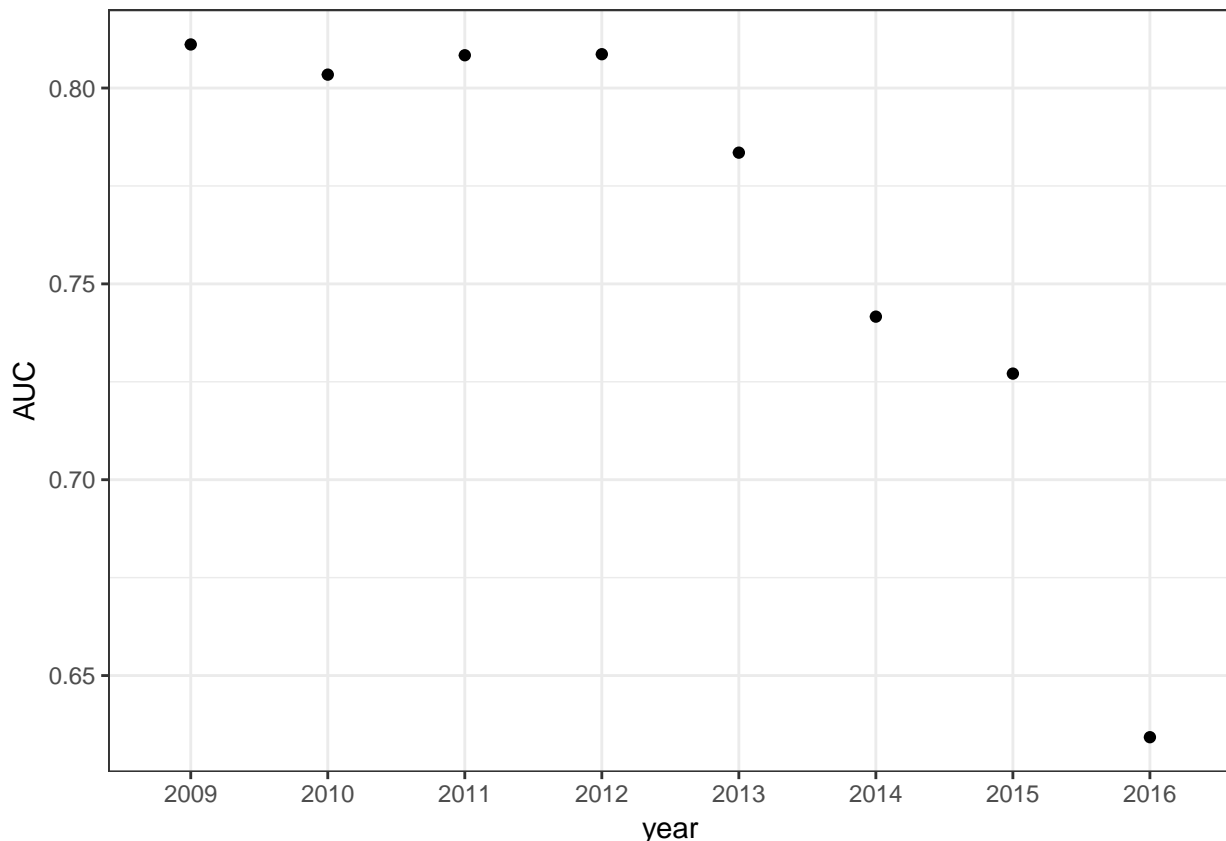
  test.predicted.probability <- predict(sqf2008_model, newdata = temp.df, type='response')
  test.pred <- prediction(test.predicted.probability, temp.df$found.weapon)
  test.perf <- performance(test.pred, "auc")
}
```

```

model2008_AUCs[which(model2008_AUCs$year == i),2] <- test.perf@y.values[[1]]
}

#Plot
theme_set(theme_bw())
p <- ggplot(data=model2008_AUCs, aes(y=AUC, x=factor(year)))
p +
  geom_point()+
  xlab("year")

```



We see the AUC steadily dropping as the years increase after 2008, which indicates that the model overfit on at least some predictors that changed in distribution after 2008— for example, predictors influenced by exogenous changes in policy and people. Although the AUC does decrease, it remains above 0.50, implying that it does better than complete randomness (no model). This may be improved if we had used more years for our training data or constructed a better model of the relationship between the predictors based on prior knowledge of the data generating process.

## Question 2C

For this question, you will generate a performance and calibration plot (like the ones created in lecture 6) for a classifier of your choice by following the steps below. You must repeat this once for each team member (e.g., if there are two people on your team, you must choose two classifiers and generate a performance and calibration plot for each). Write at least one paragraph (per classifier) explaining what you did and what you found.

## Answer 2C: Model 1 [James Wu]

```
#James read in filter to arrest.reasons
James.data <- sqf_08_16 %>% filter(grepl('marihuana|substance',suspected.crime))

#Convert variable types to factors as necessary and standardize real-value variables
James.data <- James.data %>%
  mutate(suspect.race = as.factor(suspect.race),
         suspect.build = as.factor(suspect.build),
         suspect.sex = as.factor(suspect.sex),
         location.housing = as.factor(location.housing),
         day = as.factor(day),
         month = as.factor(month),
         time.period = as.factor(time.period),
         precinct = as.factor(precinct)) %>%
  mutate(suspect.height = standardize(suspect.height),
         suspect.weight = standardize(suspect.weight),
         suspect.age = standardize(suspect.age),
         observation.period = standardize(observation.period))

#Set training data to 2008-2010
James.training <- James.data %>% filter(year == 2008:2010)

## Warning in year == 2008:2010: longer object length is not a multiple of
## shorter object length

#Set test data to 2011
James.test <- James.data %>% filter(year == 2011)

#Logistic Model
James_model <- glm(formula = found.contraband ~ 1 + precinct + location.housing + additional.report +
                  additional.investigation + additional.proximity + additional.evasive +
                  additional.associating + additional.direction + additional.highcrime + additional.
                  additional.sights + additional.other + stopped.bc.object + stopped.bc.desc +
                  stopped.bc.casing + stopped.bc.lookout + stopped.bc.clothing + stopped.bc.drugs +
                  stopped.bc.furtive + stopped.bc.violent + stopped.bc.bulge + stopped.bc.other +
                  suspect.age + suspect.sex + suspect.build + suspect.height + suspect.weight +
                  suspect.race + inside + radio.run + observation.period + day + month + time.period,
                  family = "binomial",data = James.training)

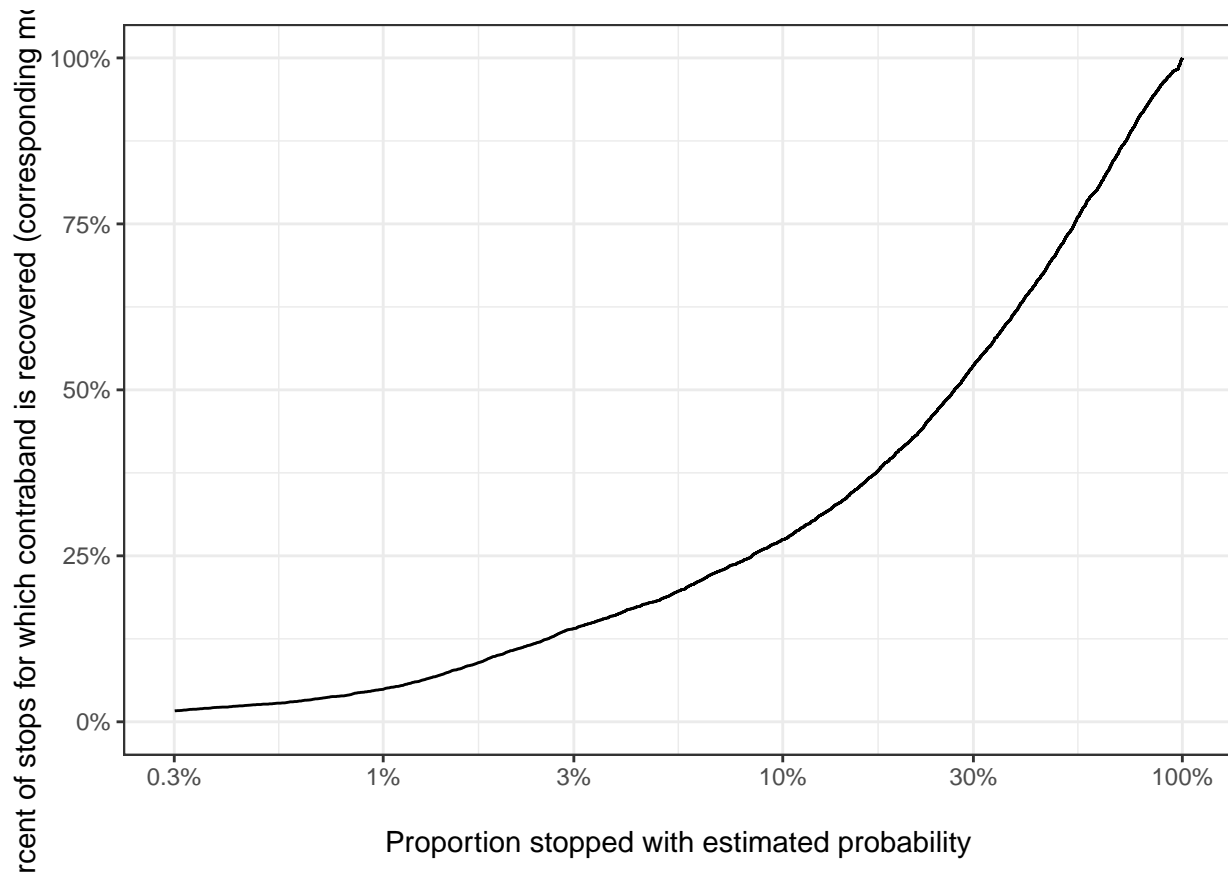
#Generate predictions for test set
James.test$predicted.probability <- predict(James_model,newdata = James.test,type = 'response')

#Performance plot
plot.data <- James.test %>% arrange(desc(predicted.probability)) %>%
  mutate(numstops = row_number(), percent.outcome = cumsum(found.contraband)/sum(found.contraband),
         stops = numstops/n()) %>% select(stops, percent.outcome)

#create plot
theme_set(theme_bw())
James.p1 <- ggplot(data=plot.data, aes(x=stops, y=percent.outcome))
James.p1 <- James.p1 + geom_line()
James.p1 <- James.p1 + scale_x_log10('\nProportion stopped with estimated probability', limits=c(0.003,
                                                labels=c('0.3%', '1%', '3%', '10%', '30%', '100%')))
```

```
James.p1 <- James.p1 + scale_y_continuous("Percent of stops for which contraband is recovered (corresponding m
James.p1
```

```
## Warning: Removed 181 rows containing missing values (geom_path).
```



```
#Calibration plot
```

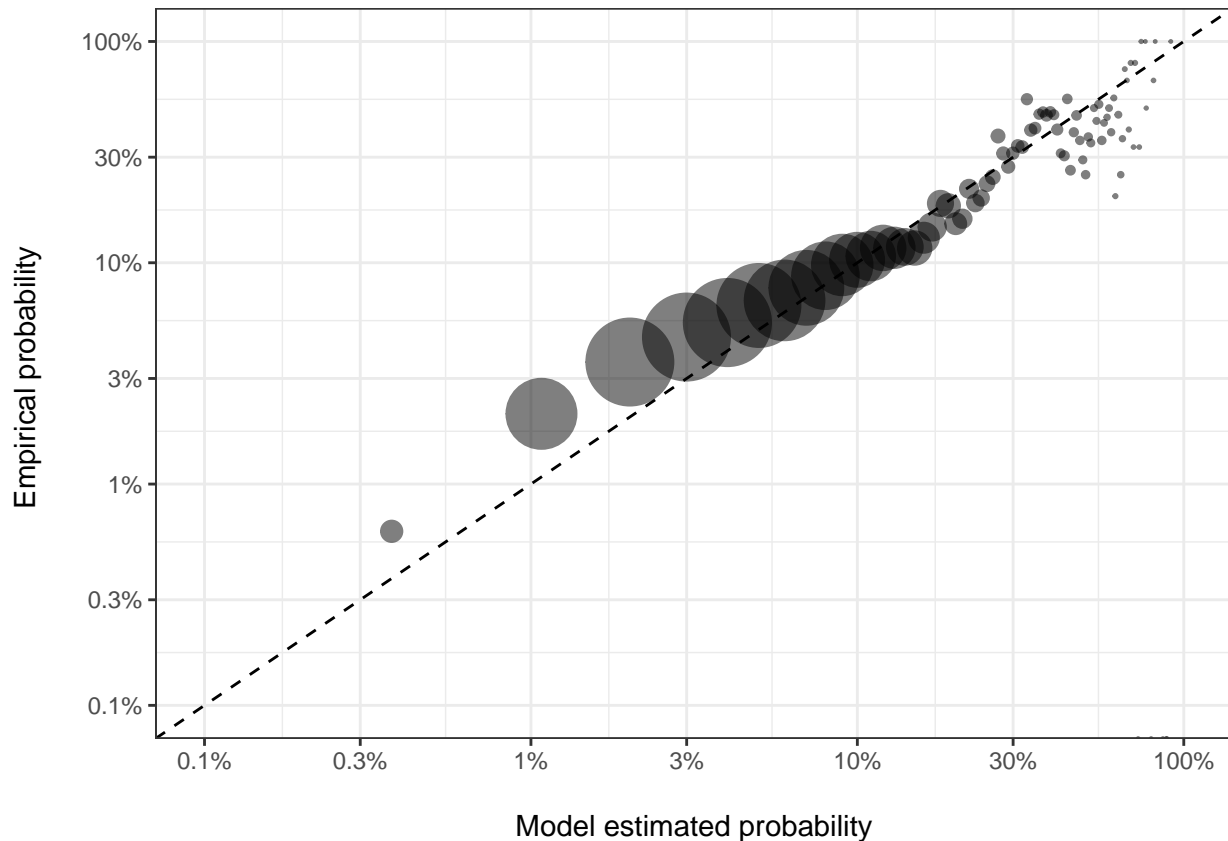
```
plot.data <- James.test %>%
  mutate(rounded.pred = round(predicted.probability,digits = 2)*100) %>%
  group_by(rounded.pred) %>%
  summarise(model.estimate = mean(predicted.probability),
            numstops = n(),
            empirical.estimate = mean(found.contraband))
```

```
#create plot
```

```
James.p2 <- ggplot(data = plot.data, aes(y = empirical.estimate, x = model.estimate))
James.p2 <- James.p2 + geom_point(aes(size = numstops), alpha = 0.5)
James.p2 <- James.p2 + scale_size_area(guide='none', max_size=15)
James.p2 <- James.p2 + geom_abline(intercept=0, slope=1, linetype="dashed")
James.p2<- James.p2 + scale_y_log10('Empirical probability \n', limits=c(.001,1), breaks=c(.001,.003,.005,.01,.03,.1,1),
                                labels=c('0.1%','0.3%','1%','3%','10%','30%','100%'))
James.p2 <- James.p2 + scale_x_log10('\nModel estimated probability', limits=c(.001,1), breaks=c(.001,.003,.005,.01,.03,.1,1),
                                labels=c('0.1%','0.3%','1%','3%','10%','30%','100%'))
James.p2
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



We decided to use `found.contraband` as our first classifier, for which we first subsetting (i.e. conditioned) the data from 2008-2010 to only include entries for suspicion of criminal possession or sale of ‘marihuana’ and ‘controlled substance’. We wanted to build a model that predicts the likelihood contraband is found given that the stop was motivated by a suspicion for suspicion of criminal possession or sale of ‘marihuana’ and ‘controlled substance’. The training set is comprised of data from 2008-2010 and the test data is 2011 data, with the assumption that going further out would make the results less accurate due to unmeasured predictors. We fitted parameters of a linear regression model on: -precinct; -whether the stop occurred in transit, housing, or on the street; -the ten additional stop circumstances (additional.); -the ten primary stop circumstances (*stopped.bc.*); -suspect age, race, build, sex, height, and weight; -whether the stop occurred inside, and whether the stop was the result of a radio call; -length of observation period; -day, month, and time of day. The performance plot tells us how many positive cases we actually ‘caught’ (recall) versus each of 100 thresholds of the predicted probabilities. The curve suggests that when our predicted probability was at a low threshold, our recall was low; in other words, we missed a lot of the positive cases. However, as probability thresholds increased, the model performed much better. The calibration plot indicates the model is mostly underestimating the actual found contrabands.

## Answer 2C: Model 2 [Lauren Broffman]

```
#Filter to only year = 2012-2014 & convert variable types to factors as necessary and standardize real-
sqf.data.12thru14 <- sqf.data %>% filter(year == 2012:2014) %>%
  mutate(suspect.race = as.factor(suspect.race),
         suspect.build = as.factor(suspect.build),
         suspect.sex = as.factor(suspect.sex),
         location.housing = as.factor(location.housing),
         day = as.factor(day),
         month = as.factor(month),
```



```

    time.period = as.factor(time.period),
    precinct = as.factor(precinct)) %>%
mutate(suspect.height = standardize(suspect.height),
       suspect.weight = standardize(suspect.weight),
       suspect.age = standardize(suspect.age),
       observation.period = standardize(observation.period))

## Warning in year == 2012:2014: longer object length is not a multiple of
## shorter object length

#Filter to only year = 2015/ This is is our test set.
sqf.data.2015 <- sqf_08_16 %>% filter(year == 2015) %>%
  mutate(suspect.race = as.factor(suspect.race),
         suspect.build = as.factor(suspect.build),
         suspect.sex = as.factor(suspect.sex),
         location.housing = as.factor(location.housing),
         day = as.factor(day),
         month = as.factor(month),
         time.period = as.factor(time.period),
         precinct = as.factor(precinct)) %>%
  mutate(suspect.height = standardize(suspect.height),
         suspect.weight = standardize(suspect.weight),
         suspect.age = standardize(suspect.age),
         observation.period = standardize(observation.period))

#build model to predict whether an officer puts individual against a wall

force.wall_model <- glm(formula = force.wall ~ location.housing + additional.report +
                        additional.investigation + additional.proximity + additional.evasive +
                        additional.associating + additional.direction + additional.highcrime + additional.
                        additional.sights + additional.other + stopped.bc.object + stopped.bc.desc +
                        stopped.bc.casing + stopped.bc.lookout + stopped.bc.clothing + stopped.bc.drugs +
                        stopped.bc.furtive + stopped.bc.violent + stopped.bc.bulge + stopped.bc.other +
                        suspect.age + suspect.sex + suspect.build + suspect.height + suspect.weight +
                        inside + radio.run + observation.period + day + month + time.period + city + suspect.family = "binomial", data = sqf.data.12thru14)

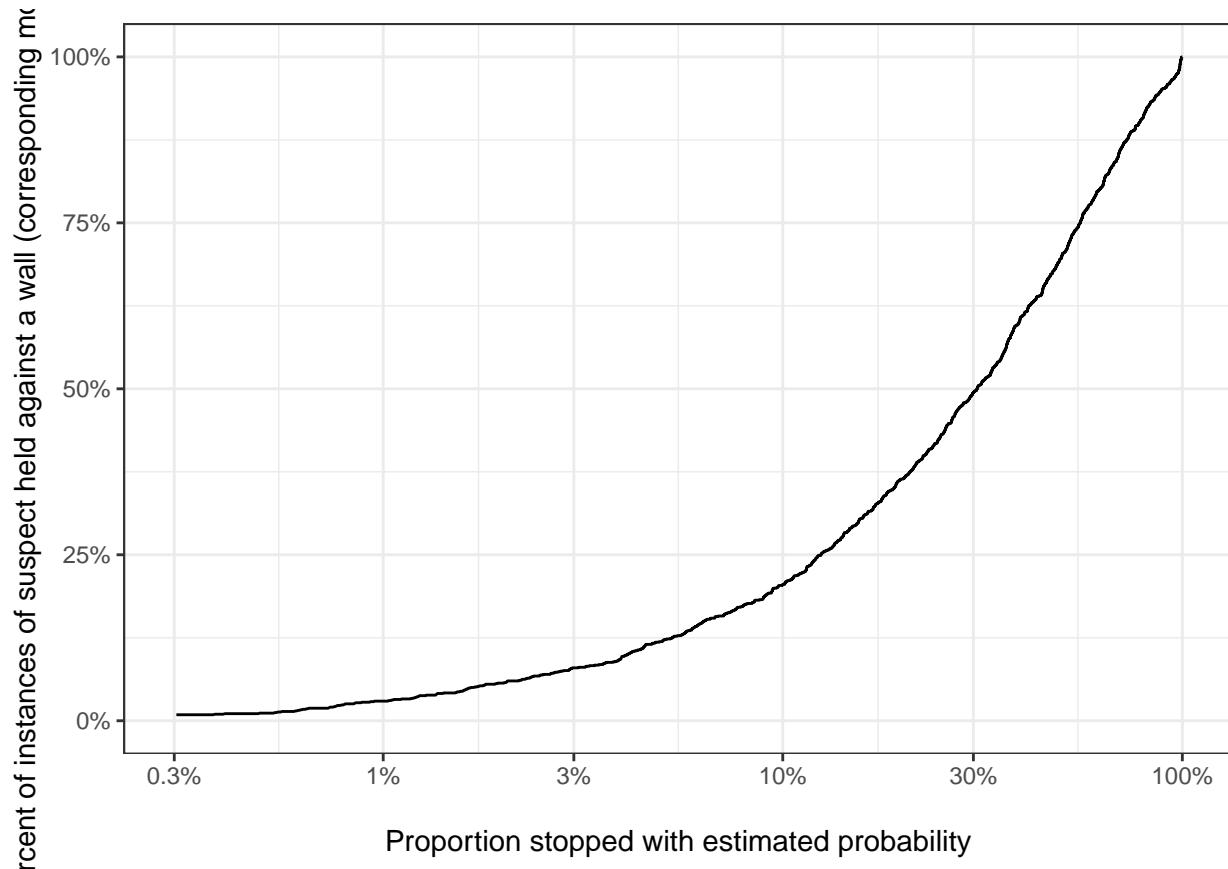
#predict probabilities
# 1. generate predictions for test set
sqf.data.2015$predicted.probability <- predict(force.wall_model, newdata = sqf.data.2015, type='response')

# make performance plot
plot.data <- sqf.data.2015 %>% arrange(desc(predicted.probability)) %>%
  mutate(numstops = row_number(), percent.outcome = cumsum(force.wall)/sum(force.wall),
         stops = numstops/n()) %>% select(stops, percent.outcome)

# create performance plot
theme_set(theme_bw())
pp <- ggplot(data=plot.data, aes(x=stops, y=percent.outcome))
pp <- pp + geom_line()
pp <- pp + scale_x_log10('\nProportion stopped with estimated probability', limits=c(0.003, 1), breaks=c(
  labels=c('0.3%', '1%', '3%', '10%', '30%', '100%'))
pp <- pp + scale_y_continuous("Percent of instances of suspect held against a wall (corresponding model
pp

```

```
## Warning: Removed 67 rows containing missing values (geom_path).
```



```
# 3) make calibration plot
```

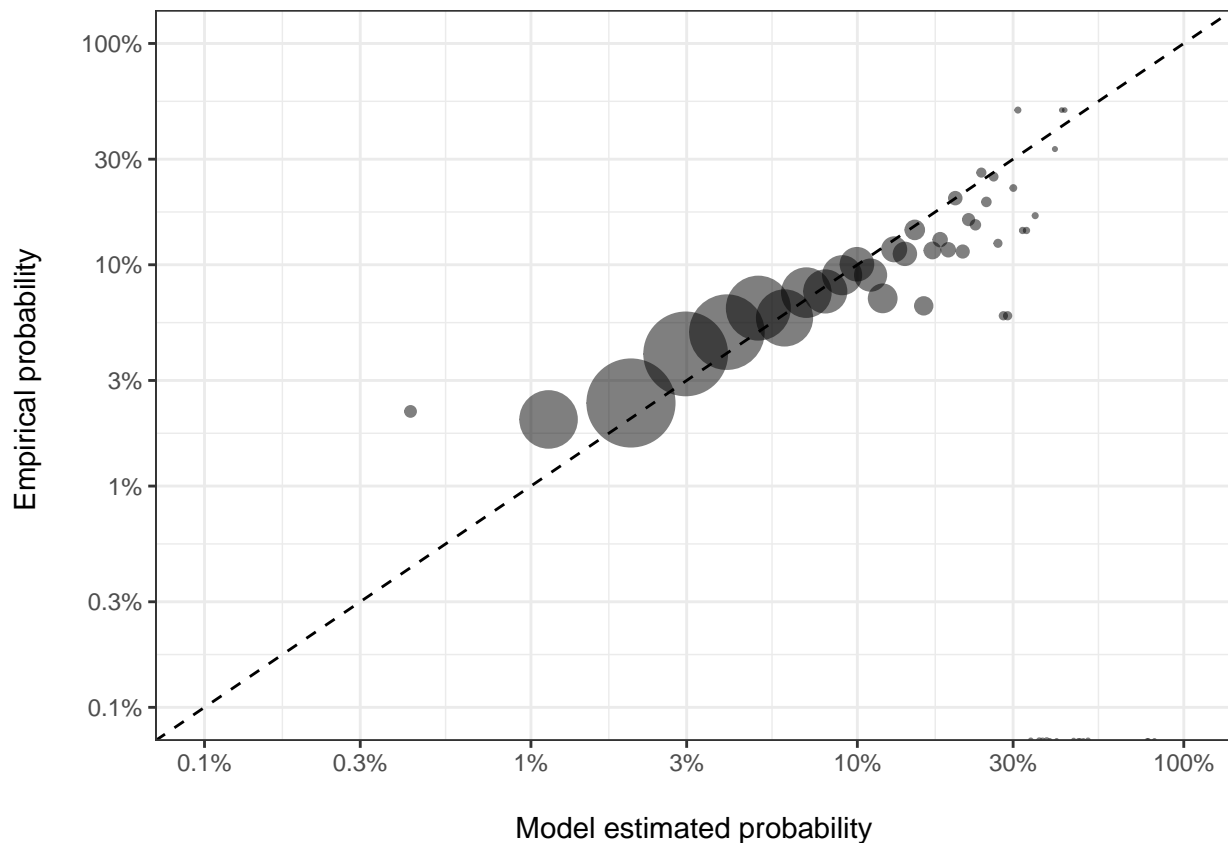
```
plot.data <- sqf.data.2015 %>% mutate(calibration = round(100*predicted.probability)) %>%
  group_by(calibration) %>% summarize(model.estimate = mean(predicted.probability),
                                     numstops = n(),
                                     empirical.estimate = mean(force.wall))
```

```
# create and save plot
```

```
cp <- ggplot(data = plot.data, aes(y=empirical.estimate, x=model.estimate))
cp <- cp + geom_point(alpha=0.5, aes(size=numstops))
cp <- cp + scale_size_area(guide='none', max_size=15)
cp <- cp + geom_abline(intercept=0, slope=1, linetype="dashed")
cp <- cp + scale_y_log10('Empirical probability \n', limits=c(.001,1), breaks=c(.001,.003,.01,.03,.1,.3,1),
                        labels=c('0.1%', '0.3%', '1%', '3%', '10%', '30%', '100%'))
cp <- cp + scale_x_log10('\nModel estimated probability', limits=c(.001,1), breaks=c(.001,.003,.01,.03,.1,.3,1),
                        labels=c('0.1%', '0.3%', '1%', '3%', '10%', '30%', '100%'))
cp
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Using whether or not the individual was subject to the specific kind use force described #as being held up against a wall as a classifier, we used logistic regression and a set of covariates (very similar to the original model, but adding in variables denoting borough and race) on a set of training data using years 2012-2014 to estimate a function by which to predict the probability that a person was subject to a specific use of force, and tested out this function on a test dataset using year 2015, to evaluate the model. First, it is worth noting that the actual instances of this kind of force being used is very low (across all the data). The performance plot is almost identical to that in the plot for the first model. Again, the performance plot tells us how many positive cases we actually ‘caught’ (recall) over all the thresholds of the predicted probabilities. Just like with the first model, the curve suggests that when our predicted probability was at a low threshold, our recall was low; in other words, we missed a lot of the positive cases. However, as probability thresholds increased, the model performed much better. The calibration plot compares the model probability to the actual probability - here, most points do not fall on the 45 degree line, as they would if the model were doing a better job predicting actual instances of suspects forced against a wall after a stop. Most of the points are above the line, indicating that the model underestimates the empirical probability of this occurrence.

## Answer 2C: Model 3 [Ravi B. Sojitra]

```
# Prep data for modeling (convert to factors and standardize)
ravi.data <- sqf_08_16 %>%
  filter (is.na(arrested) == FALSE) %>%
  mutate(suspect.race = as.factor(suspect.race), # Factors.
         suspect.build = as.factor(suspect.build),
         suspect.sex = as.factor(suspect.sex),
         location.housing = as.factor(location.housing),
         day = as.factor(day),
```

```

        month = as.factor(month),
        time.period = as.factor(time.period),
        precinct = as.factor(precinct)) %>%
mutate(suspect.height = standardize(suspect.height), # Standardize.
       suspect.weight = standardize(suspect.weight),
       suspect.age = standardize(suspect.age),
       observation.period = standardize(observation.period))

training.set <- ravi.data %>% filter(year %in% 2008:2010)
testing.set <- ravi.data %>% filter(year %in% 2011)

ravi.model <- glm(formula = arrested ~
  1 + precinct + location.housing + additional.report +
  additional.investigation + additional.proximity + additional.evasive +
  additional.associating + additional.direction + additional.highcrime
  + additional.time +
  additional.sights + additional.other + stopped.bc.object + stopped.bc.desc +
  stopped.bc.casing + stopped.bc.lookout + stopped.bc.clothing + stopped.bc.drugs +
  stopped.bc.furtive + stopped.bc.violent + stopped.bc.bulge + stopped.bc.other +
  suspect.age + suspect.sex + suspect.build + suspect.height + suspect.weight +
  suspect.race + suspect.hair + inside + radio.run + observation.period + day + month,
  family = "binomial", data = training.set)

ticks = c(.003,.01,.03,.1,.3,1)
ticklabels = c('0.3%', '1%', '3%', '10%', '30%', '100%')

performance.xlabel <- '\nProportion stopped with estimated probability'
performance.ylabel <- 'Percent Arrested (i.e. corresponding model recall)'
calibration.xlabel <- '\nModel estimated probability'
calibration.ylabel <- 'Empirical probability \n'

testing.set$predictedps <- predict(ravi.model, newdata = testing.set, type='response')

ravi.performance.data <- testing.set %>%
  arrange(desc(predictedps)) %>%
  mutate(numstops = row_number(),
         percent.outcome = cumsum(arrested)/sum(arrested),
         stops = numstops/n()
  ) %>%
  select(stops, percent.outcome)
ravi.calibration.data <- testing.set %>%
  mutate(calibration=round(100 * predictedps)) %>%
  group_by(calibration) %>%
  summarize(model.estimate = mean(predictedps),
            numstops = n(),
            empirical.estimate = mean(arrested))

theme_set(theme_bw())
ravi.performance.plot <- ggplot(data=ravi.performance.data,
                               aes(x=stops, y=percent.outcome))
ravi.performance.plot <- ravi.performance.plot + geom_line()
ravi.performance.plot <- ravi.performance.plot + scale_x_log10(performance.xlabel,
                                                                limits=c(0.003, 1),

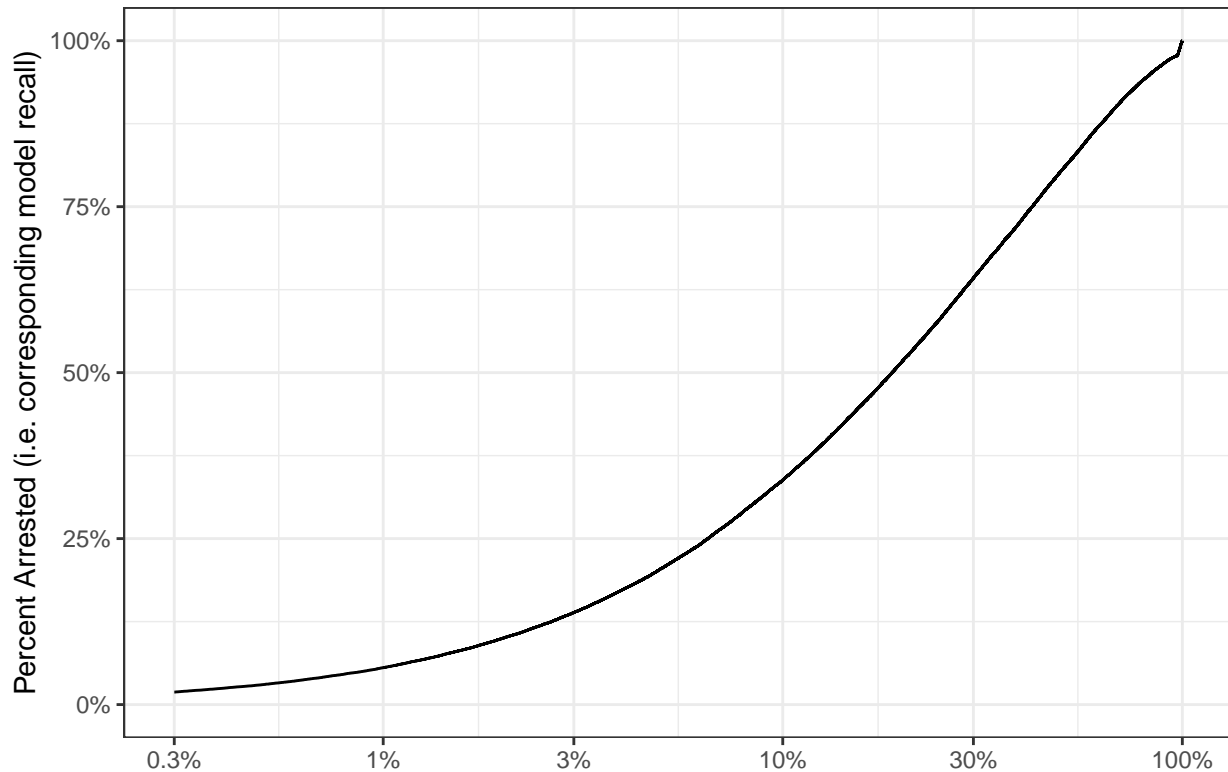
```

```

                                breaks=ticks,
                                labels=ticklabels)
ravi.performance.plot <- ravi.performance.plot + scale_y_continuous(performance.ylabel,
                                                                    limits=c(0, 1), labels=scales::percent)
ravi.performance.plot

```

## Warning: Removed 2046 rows containing missing values (geom\_path).



Proportion stopped with estimated probability

```

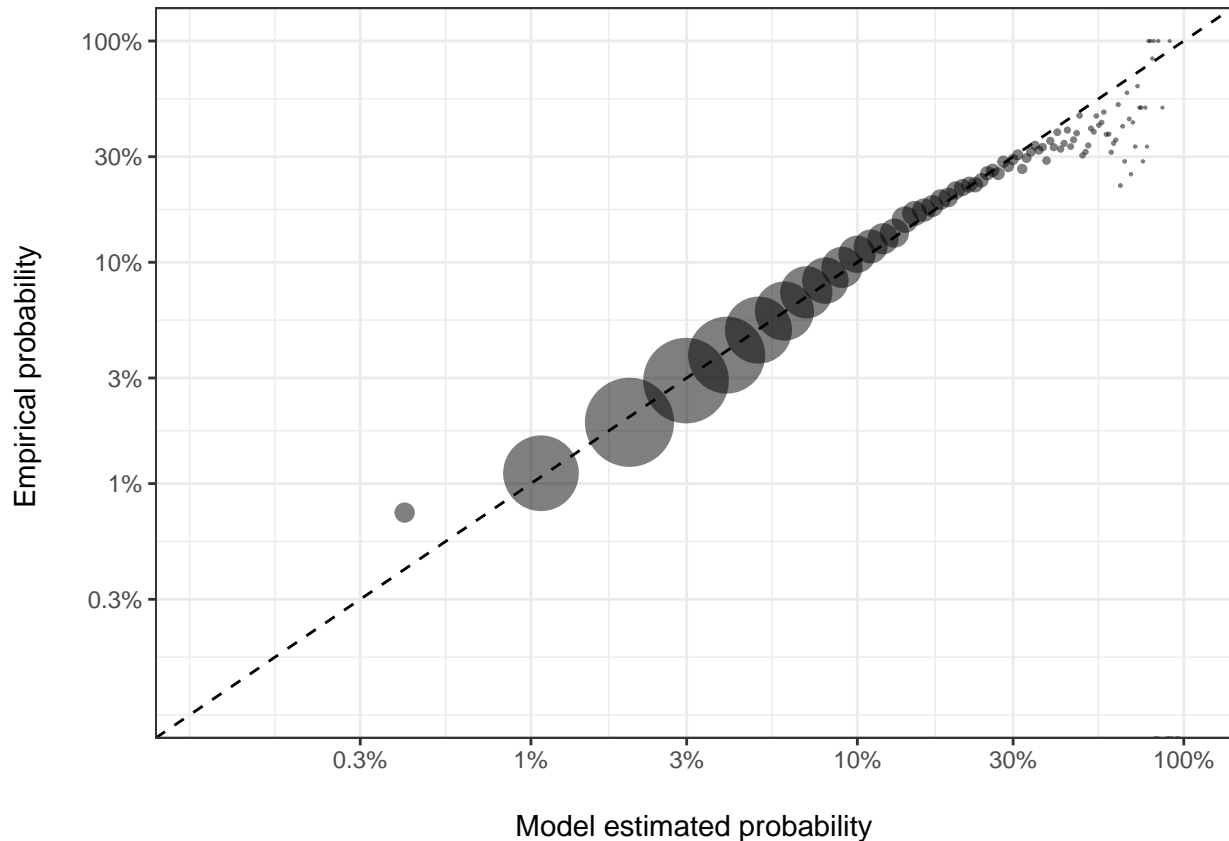
# create and save plot
ravi.calibration.plot <- ggplot(data = ravi.calibration.data,
                                aes(y=empirical.estimate, x=model.estimate))
ravi.calibration.plot <- ravi.calibration.plot +
  geom_point(alpha=0.5, aes(size=numstops))
ravi.calibration.plot <- ravi.calibration.plot +
  scale_size_area(guide='none', max_size=15)
ravi.calibration.plot <- ravi.calibration.plot +
  geom_abline(intercept=0, slope=1, linetype="dashed")
ravi.calibration.plot <- ravi.calibration.plot +
  scale_y_log10(calibration.ylabel,
                limits=c(.001,1),
                breaks=ticks,
                labels=ticklabels)
ravi.calibration.plot <- ravi.calibration.plot +
  scale_x_log10(calibration.xlabel,
                limits=c(.001,1),
                breaks=ticks,
                labels=ticklabels)

```

```
ravi.calibration.plot
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



For our third classifier, we used a logistic regression model to predict the likelihood of arrest. Using predictors similar to what we used in class as well as others such as hair color, we used various continuous and categorical variables to perform the classification. Because we are predicting whether an arrest is made, there are more variables that can be used as predictors compared to response variables that are about events that happen before the arrest (e.g. if we were to predict an individual's stopped, we cannot use the `stopped.bc*` variables because that information is collected after the fact), which may be potentially useful for training a better model. For the train and test split, we trained the model on data from 2008 to 2010 and tested the model on data from 2011. Plotting the performance reveals that following the model's recommendation may yield a large rate of arrests per percent increase in model certainty, at low thresholds; however, in general the recall is pretty poor (i.e. at low thresholds, people are more likely not to be arrested given they are stopped). Compared to the first two models' calibration plots, this calibration plot reveals that the model certainty is better aligned with the empirical probabilities of arrest, since most of the data lies on the slope = 1 line. However, it is important to note that the model does overestimate certainty at high thresholds and underestimates certainty at low thresholds. most of the points do not sit exactly on the slope = 1 line.