ACT03 - Data Mining

Team 7 | Captain: Nadine Rose | Members: Kyle Scott, Lakshya Rathore, Bailey LaRea

February 8, 2022

Problem 1.1

```r
library(kableExtra)

Observation <- c(1:12)
True_Status <- c(1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
Posterior_Probability <- c(0.95, 0.85, 0.75, 0.45, 0.35, 0.25, 0.15, 0.05, 0.65, 0.55, 0.5, 0.7)

df <- data.frame(Observation, True_Status, Posterior_Probability)

kable(df, linesep = "\\addlinespace",
      digits = 3, booktabs = T, format = 'latex') %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Observation	True_Status	Posterior_Probability
1	1	0.95
2	1	0.85
3	1	0.75
4	1	0.45
5	1	0.35
6	1	0.25
7	0	0.15
8	0	0.05
9	0	0.65
10	0	0.55
11	0	0.50
12	0	0.70

We are given that the cut-off point is 0.72. Using this chart, we know that any Posterior Probability ≥ 0.72 will be classified as 1 and any Posterior Probability < 0.72 will be classified as 0.

```r
Predicted_Status <- c(1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
df <- cbind(df, Predicted_Status)


kable(df, linesep = "\\addlinespace",
      digits = 3, booktabs = T, format = 'latex') %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Observation	True_Status	Posterior_Probability	Predicted_Status
1	1	0.95	1
2	1	0.85	1
3	1	0.75	1
4	1	0.45	0
5	1	0.35	0
6	1	0.25	0
7	0	0.15	0
8	0	0.05	0
9	0	0.65	0
10	0	0.55	0
11	0	0.50	0
12	0	0.70	0

Now we can compare True_Status to Predicted_Status to determine the number of True Positives, False Positives, True Negatives and False Negatives.

Answer Section:

True Positive (True_Status = 1 && Predicted_Status = 1) = 3 (Obs. 1, 2, and 3)

False Positive (True_Status = 0 && Predicted_Status = 1) = 0

True Negative (True_Status = 0 && Predicted_Status = 0) = 6 (Obs. 7, 8, 9, 10, 11, and 12)

False Negative (True_Status = 1 && Predicted_Status = 0) = 3 (Obs. 4, 5, and 6)

Sensitivity $= \frac{N_{TP}}{N_{TP}+N_{FN}} = \frac{3}{3+3} = 0.5$

Specificity $= \frac{N_{TN}}{N_{TN}+N_{FP}} = \frac{6}{6+0} = 1$

Accuracy $= \frac{N_{TP}+N_{TN}}{N} = \frac{3+6}{12} = 0.75$

Precision $= \frac{N_TP}{N_{TP}+N_{FP}} = \frac{3}{3+0} = 1$

F1 Score $= \frac{2\times(\text{Precision}\times\text{Sensitivity})}{\text{Precision}+\text{Sensitivity}} = \frac{2\times1\times0.5}{0.5+1} = 0.667$

Problem 1.2

```
Rank <- c(12, 11, 10, 5, 4, 3, 2, 1, 8, 7, 6, 9)

Rank_Positive <- c(12, 11, 10, 5, 4, 3, 0, 0, 0, 0, 0, 0)

df <- cbind(df, Rank)

df <- cbind(df, Rank_Positive)

kable(df, linesep = "\\addlinespace",

      digits = 3, booktabs = T, format = 'latex') %>%

  kable_styling(latex_options = c("striped", "hold_position"))
```

Observation	True_Status	Posterior_Probability	Predicted_Status	Rank	Rank_Positive
1	1	0.95	1	12	12
2	1	0.85	1	11	11
3	1	0.75	1	10	10
4	1	0.45	0	5	5
5	1	0.35	0	4	4
6	1	0.25	0	3	3
7	0	0.15	0	2	0
8	0	0.05	0	1	0
9	0	0.65	0	8	0
10	0	0.55	0	7	0
11	0	0.50	0	6	0
12	0	0.70	0	9	0

```
sum_rankpos <- sum(Rank_Positive)

sum_rankpos
```

```
## [1] 45
```

AUC $= \frac{\text{sumrankpos - } 0.5 \times \pi N \times (\pi N + 1)}{\pi N (N - \pi N)} = \frac{45 - 0.5 \times 0.5 \times 12 \times (0.5 \times 12 + 1)}{0.5 \times 12 \times (12 - 0.5 \times 12)} = 0.667$

Problem 1.3

$\text{GINI} = 2 \times (\text{AUC} - 0.5) = 2 \times (0.667 - 0.5) = 0.3334$

Problem 2

Calculate KS Statistics

```r
# create columns for Decile, Positive, and Negative as given in the Assignment
Decile <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
Positive <- c(100, 98, 96, 90, 85, 80, 75, 66, 51, 41)
Negative <- c(0, 2, 4, 10, 15, 20, 25, 34, 49, 59)

# create a data frame with these 3 columns
ks <- data.frame(Decile, Positive, Negative)
```

```r
# create vars for Total Num of Positives and Negatives
total_pos <- sum(Positive)
total_neg <- sum(Negative)

# calculate TPR
TPR <- Positive / total_pos

# calculate TNR
TNR <- Negative / total_neg

# calculate cumulative TPR
cumulative_TPR <- cumsum(TPR)

# calculate cumulative TNR
cumulative_TNR <- cumsum(TNR)
```

```r
# calculate KS statistic
KS <- cumulative_TPR - cumulative_TNR
```

```r
# bind TPR to the data frame
ks <- cbind(ks, TPR)


# bind TNR to the data frame
ks <- cbind(ks, TNR)


# bind cumulative_TPR to the data frame
ks <- cbind(ks, cumulative_TPR)


# bind cumulative_TNR to the data frame
ks <- cbind(ks, cumulative_TNR)


# bind KS to the data frame
ks <- cbind(ks, KS)
```

```r
library(kableExtra)


kable(ks, caption = "Calculating KS Statistics", linesep = "\\addlinespace",
      digits = 3, booktabs = T, format = 'latex') %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: Calculating KS Statistics

Decile	Positive	Negative	TPR	TNR	cumulative_TPR	cumulative_TNR	KS
1	100	0	0.128	0.000	0.128	0.000	0.128
2	98	2	0.125	0.009	0.253	0.009	0.244
3	96	4	0.123	0.018	0.376	0.028	0.348
4	90	10	0.115	0.046	0.491	0.073	0.418
5	85	15	0.109	0.069	0.600	0.142	0.458
6	80	20	0.102	0.092	0.702	0.234	0.468
7	75	25	0.096	0.115	0.798	0.349	0.449
8	66	34	0.084	0.156	0.882	0.505	0.378
9	51	49	0.065	0.225	0.948	0.729	0.218
10	41	59	0.052	0.271	1.000	1.000	0.000

Problem 3.1

Read in the CSV file for Microsoft

```
library(readr)


microsoft <- read.csv("Microsoft_Results.csv")
```

Problem 3.2

First Step is to create a function that will calculate all the required values which can be passed to the function parameter called 'cutoff'

```
stats <- function(cutoff){

predicted_detections <- ifelse(microsoft$P_HasDetections >= cutoff, 1, ifelse(microsoft$P_HasDetections <

TP <- sum(ifelse((microsoft$HasDetections == 1 & predicted_detections == 1), 1, 0))

TruePos <<- c(TruePos, TP)

FP <- sum(ifelse((microsoft$HasDetections == 0 & predicted_detections == 1), 1, 0))

FalsePos <<- c(FalsePos, FP)

TN <- sum(ifelse((microsoft$HasDetections == 0 & predicted_detections == 0), 1, 0))

TrueNeg <<- c(TrueNeg, TN)

FN <- sum(ifelse((microsoft$HasDetections == 1 & predicted_detections == 0), 1, 0))

FalseNeg <<- c(FalseNeg, FN)

Sensitivity <<- c(Sensitivity, TP / (TP + FN))
```

```r
Specificity <<- c(Specificity, TN / (TN + FP))


Accuracy <<- c(Accuracy, (TP + TN) / (TP + TN + FP + FN))


Precision <<- c(Precision, TP / (TP + FP))
}
```

Now that the function is created, we can call the function for whichever desired values we have. For the purpose of simplicity and demonstration, here we have chosen to display a table with all the desired statistics at the five percent level

```r
TP <- vector()
FP <- vector()
TN <- vector()
FN <- vector()
TruePos <- vector()
FalsePos <- vector()
TrueNeg <- vector()
FalseNeg <- vector()
Sensitivity <- vector()
Specificity <- vector()
Accuracy <- vector()
Precision <- vector()


cutoff <- seq(0.05, 1, by = 0.05)
mapply(stats, cutoff)
```

```r
stats_df <- data.frame(cutoff, TruePos, FalsePos, TrueNeg, FalseNeg, Sensitivity, Specificity, Accuracy, Pr


kable(stats_df, caption = "Statistics for the five percentile level", linesep = "\\addlinespace",
      digits = 3, booktabs = T, format = 'latex') %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 2: Statistics for the five percentile level

cutoff	TruePos	FalsePos	TrueNeg	FalseNeg	Sensitivity	Specificity	Accuracy	Precision
0.05	499001	500999	0	0	1.000	0.000	0.499	0.499
0.10	498968	500789	210	33	1.000	0.000	0.499	0.499
0.15	498711	498998	2001	290	0.999	0.004	0.501	0.500
0.20	497610	492899	8100	1391	0.997	0.016	0.506	0.502
0.25	492989	474524	26475	6012	0.988	0.053	0.519	0.510
0.30	478925	432025	68974	20076	0.960	0.138	0.548	0.526
0.35	451667	370007	130992	47334	0.905	0.261	0.583	0.550
0.40	413742	303093	197906	85259	0.829	0.395	0.612	0.577
0.45	367764	239301	261698	131237	0.737	0.522	0.629	0.606
0.50	306004	170809	330190	192997	0.613	0.659	0.636	0.642
0.55	225532	99644	401355	273469	0.452	0.801	0.627	0.694
0.60	156745	51740	449259	342256	0.314	0.897	0.606	0.752
0.65	111830	27026	473973	387171	0.224	0.946	0.586	0.805
0.70	84108	14959	486040	414893	0.169	0.970	0.570	0.849
0.75	66583	9018	491981	432418	0.133	0.982	0.559	0.881
0.80	51056	5406	495593	447945	0.102	0.989	0.547	0.904
0.85	33312	2738	498261	465689	0.067	0.995	0.532	0.924
0.90	3961	248	500751	495040	0.008	1.000	0.505	0.941
0.95	3	0	500999	498998	0.000	1.000	0.501	1.000
1.00	0	0	500999	499001	0.000	1.000	0.501	NaN

Problem 3.3

```r
library(pROC)


AUC <- auc(microsoft$HasDetections, microsoft$P_HasDetections)

AUC
```

```
## Area under the curve: 0.6938
```

```r
GINI <- 2*(AUC -  0.5)

GINI
```

```
## [1] 0.3875789
```

Problem 3.4

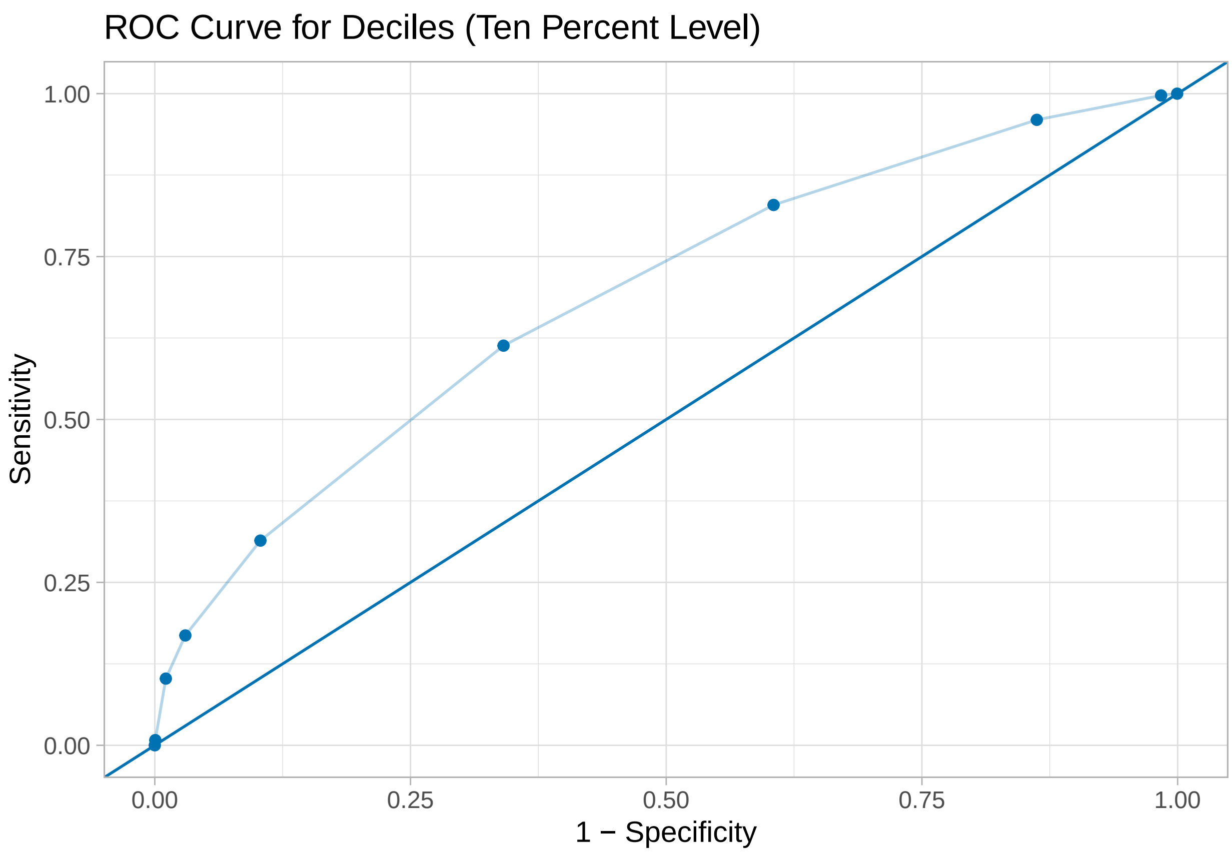
Call our function from 3.2 but this time only for deciles

```
TP <- vector()
FP <- vector()
TN <- vector()
FN <- vector()
TruePos <- vector()
FalsePos <- vector()
TrueNeg <- vector()
FalseNeg <- vector()
Sensitivity <- vector()
Specificity <- vector()
Accuracy <- vector()
Precision <- vector()

cutoff <- seq(0.1, 1, by = 0.1)
mapply(stats, cutoff)
```

Create the ROC plot by graphing 1 - Specificity vs Sensitivity with some of the data provided by our function

```
library(ggplot2)
Specificity <- 1 - Specificity
df <- data.frame(Specificity, Sensitivity)
plt <- ggplot(data = df, aes(x = Specificity, y = Sensitivity)) +
  geom_point(color = '#0072B2') +
```

```
  geom_line(alpha = 0.3, color = '#0072B2') +
  geom_abline(slope = 1, inctercept = 0, color = '#0072B2') +
  ggtitle(label = "ROC Curve for Deciles (Ten Percent Level)") +
  xlab("1 - Specificity") +
  ylab("Sensitivity") +
  theme_light()

plt
```



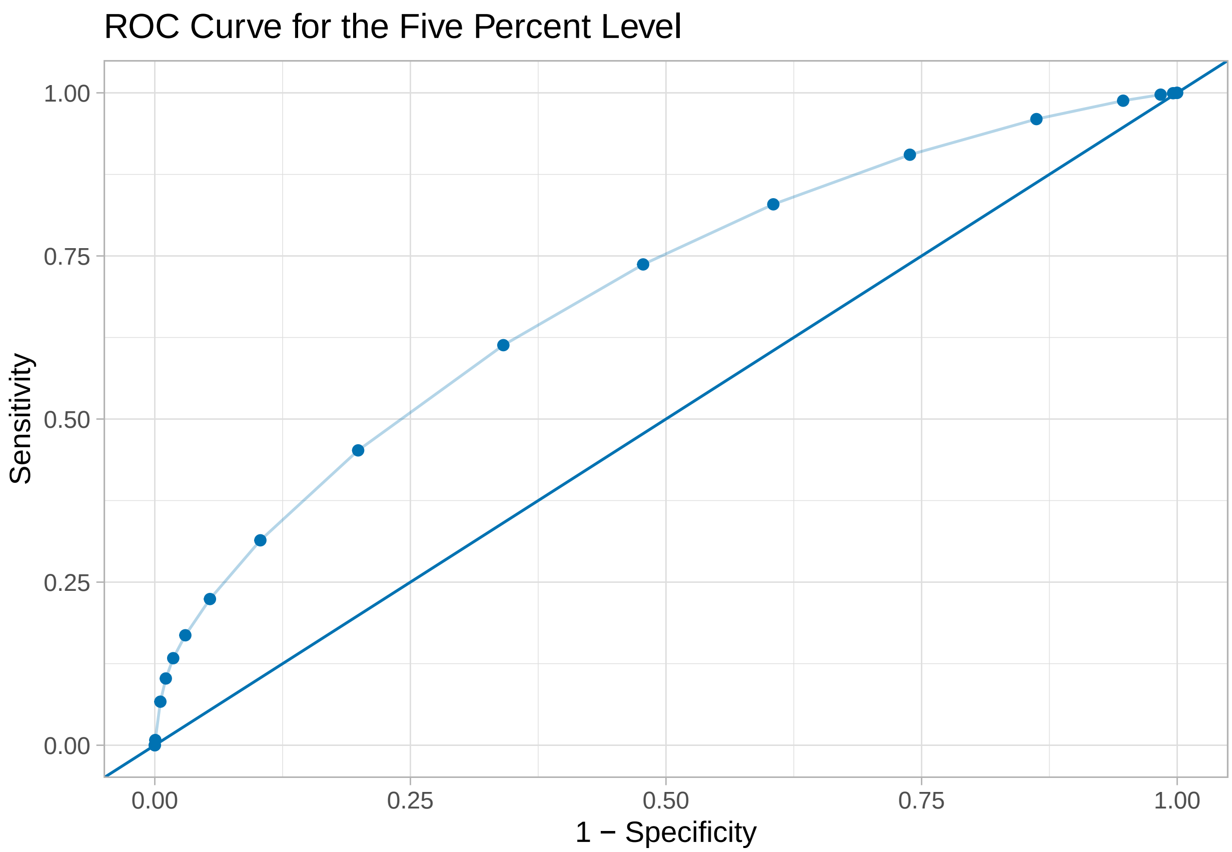
Problem 3.5

Same process as 3.4 except this time we are working at the fifth percentile level

```
TP <- vector()
FP <- vector()
TN <- vector()
FN <- vector()
TruePos <- vector()
FalsePos <- vector()
TrueNeg <- vector()
FalseNeg <- vector()
Sensitivity <- vector()
Specificity <- vector()
Accuracy <- vector()
Precision <- vector()

cutoff <- seq(0.05, 1, by = 0.05)
mapply(stats, cutoff)
```

```
library(ggplot2)
Specificity <- 1 - Specificity
df <- data.frame(Specificity, Sensitivity)
plt <- ggplot(data = df, aes(x = Specificity, y = Sensitivity)) +
  geom_point(color = '#0072B2') +
  geom_line(alpha = 0.3, color = '#0072B2') +
  geom_abline(slope = 1, inctercept = 0, color = '#0072B2') +
  ggtitle(label = "ROC Curve for the Five Percent Level") +
```

```
  xlab("1 - Specificity") +
  ylab("Sensitivity") +
  theme_light()


plt
```

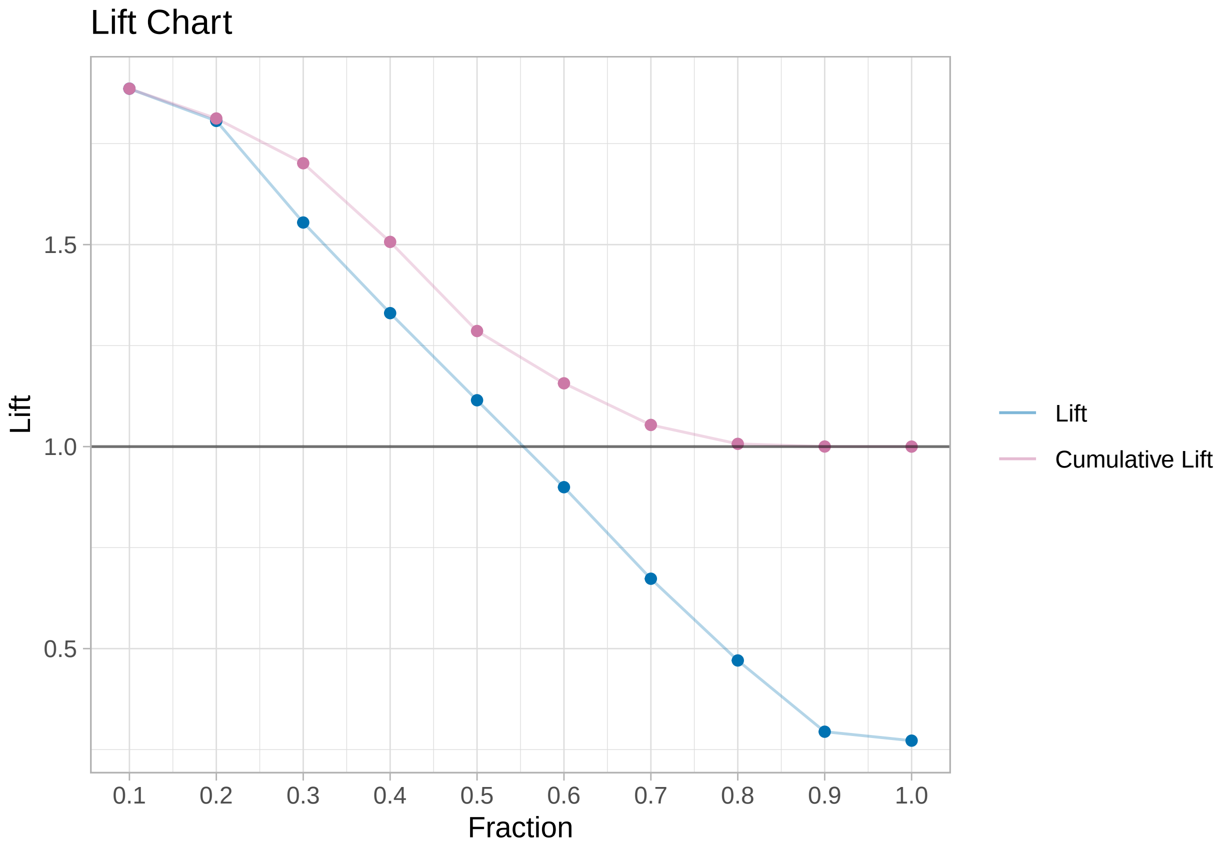


Problem 3.6

```r
true_pos <- function(decile){

vec <- microsoft$HasDetections[microsoft$P_HasDetections > decile - 0.1 & microsoft$P_HasDetections <= dec

n <- length(vec)

count <- sum(vec == 1)

TPC <<- c(TPC, count)

N <<- c(N, n)
}
```

```r
total_pos <- sum(microsoft$HasDetections == 1)
prop_pos <- total_pos / 1000000

vec <- vector()
n <- vector()
count <- vector()
TPC <- vector()
N <- vector()

decile <- seq(1, 0.1, -0.1)

mapply(true_pos, decile)
```

```r
library(ggplot2)

lift <- (TPC / N) / prop_pos

new_seq <- seq(0.1, 1, 0.1)

lift_cum <- (cumsum(TPC) / cumsum(N)) / prop_pos

df <- data.frame(lift, lift_cum, new_seq)

plt <- ggplot(data = df, aes(x = new_seq)) +
  geom_point(aes(y = lift), color = '#0072B2') +
  geom_line(aes(y = lift, color = "Lift"), alpha = 0.3) +
  geom_point(aes(y = lift_cum), color = '#CC79A7') +
  geom_line(aes(y = lift_cum, color = "Cumulative Lift"), alpha = 0.3) +
  geom_hline(yintercept = 1, alpha = 0.5) +
  scale_color_manual("", breaks = c("Lift", "Cumulative Lift"), values = c('#0072B2', '#CC79A7')) +
  ggtitle(label = "Lift Chart") +
  xlab("Fraction") +
  ylab("Lift") +
  scale_x_continuous(n.breaks = 10) +
  theme_light()

plt
```


Lift Chart
Lift
1.5
1.0
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Fraction
Lift
Cumulative Lift

Chapter IV Exercises

Problem 1

$$(4.2)\quad p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

So, $\dfrac{p(x)}{1 - p(x)}$

$$= \frac{\dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}{1 - \dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}$$

$$= \frac{\dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}{\dfrac{1 + e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} - \dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}$$

$$= \frac{\dfrac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}}{\dfrac{1}{1 + e^{\beta_0 + \beta_1 X}}}$$

$$(4.3)\quad \frac{p(X)}{1 - p(X)} = e^{\beta_0 + \beta X}$$

Problem 9

(a) (i) $\frac{p(X)}{1-p(X)} = 0.37$

(ii) $p(X) = 0.37(1 - p(X))$

(iii) $1.37 \times p(X) = 0.37$

(iv) $p(X) = 0.37/1.37 = 27\%$

(b) Odds $= \frac{p(X)}{1-p(X)} = 0.16/0.84 = 0.19$