Announcement

Homework 12

Neal Kar (ink2105)

Announcement

Please do not add code folding (code_folding: hide) to your YAML Header or echo = FALSE to your RMD code chunk options. In order to accurately grade your HTML files we need to be able to see all of your code. Thank you!

Question 1

a)

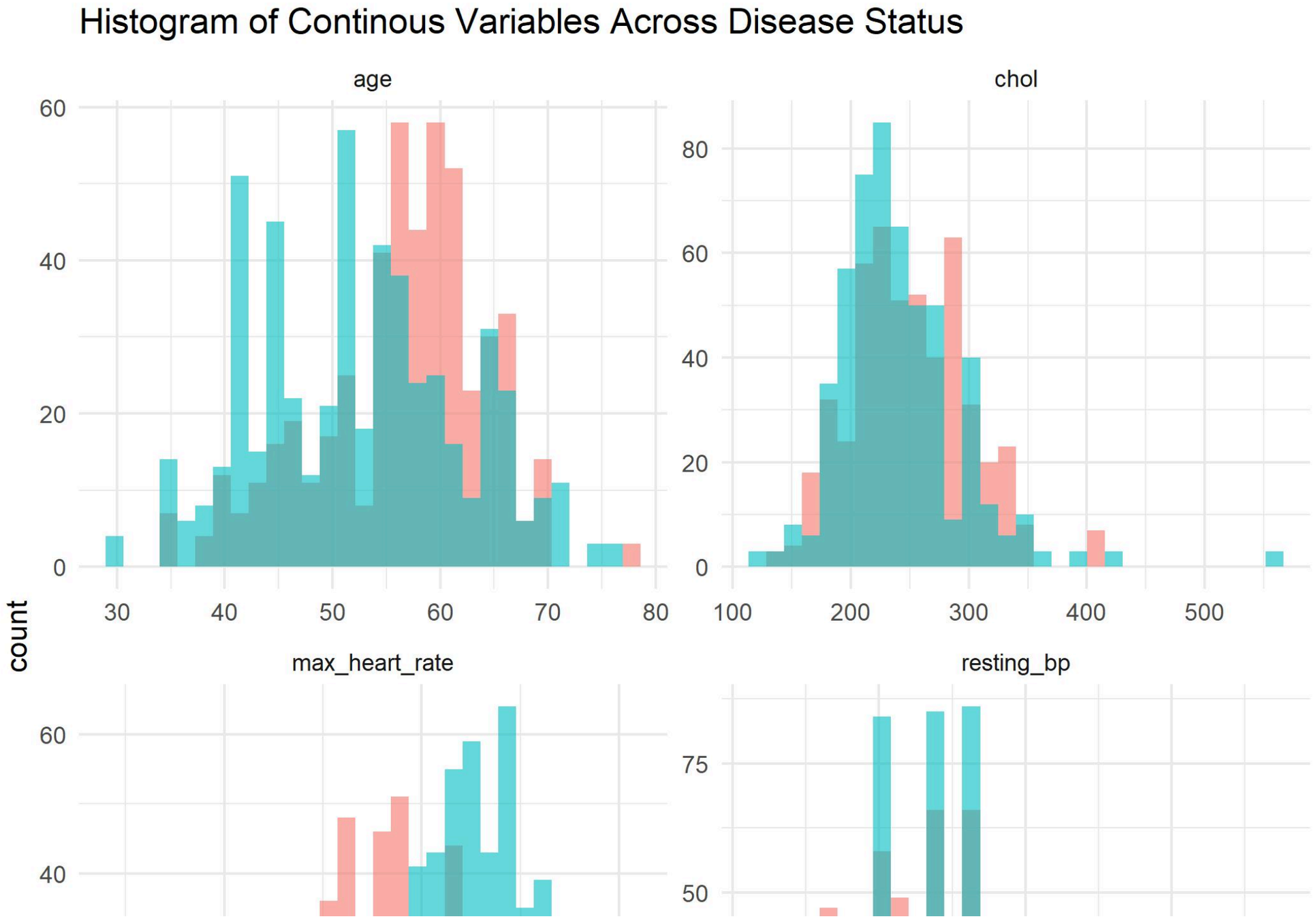
Read in the data in `heart_clean.csv`. The data contains various predictors of heart disease and an indicator variable of whether the disease is present. Information on the predictors can be found at https://www.kaggle.com/johnsmith88/heart-disease-dataset (https://www.kaggle.com/johnsmith88/heart-disease-dataset). Additionally, change `disease` to a factor variable.

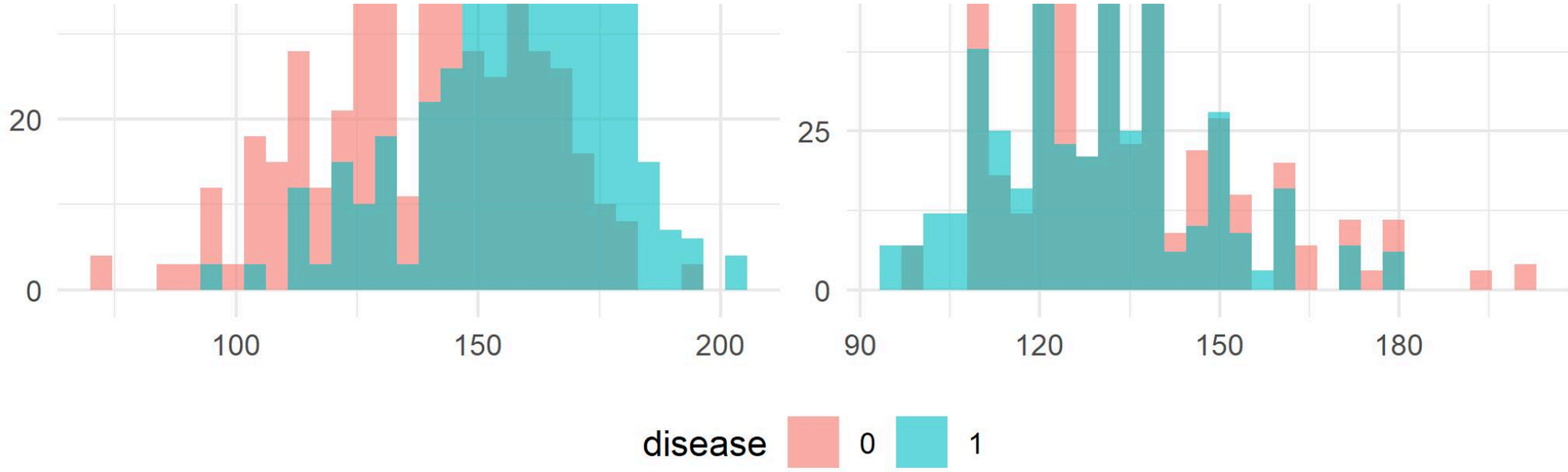
```
cvd_df_wide <- read_csv("data/heart_clean.csv")

cvd_df_wide <- cvd_df_wide %>%
  mutate(disease = factor(disease))

is.factor(cvd_df_wide$disease)
```

```
## [1] TRUE
```

b)

We wish to visualize the distribution of the variables `age` , `chol` , `max_heart_rate` , *and* `resting_bp` *across disease status. Recreate the plot below. Use (1) set position = “identity” and alpha = 0.6 in geom_histogram to create the overlapping histograms and (2) set the theme using theme_minimal().*


Histogram of Continous Variables Across Disease Status
age
chol
max_heart_rate
resting_bp
count
60
40
20
0
30
40
50
60
70
80
80
60
40
20
0
100
200
300
400
500
60
40
75
50

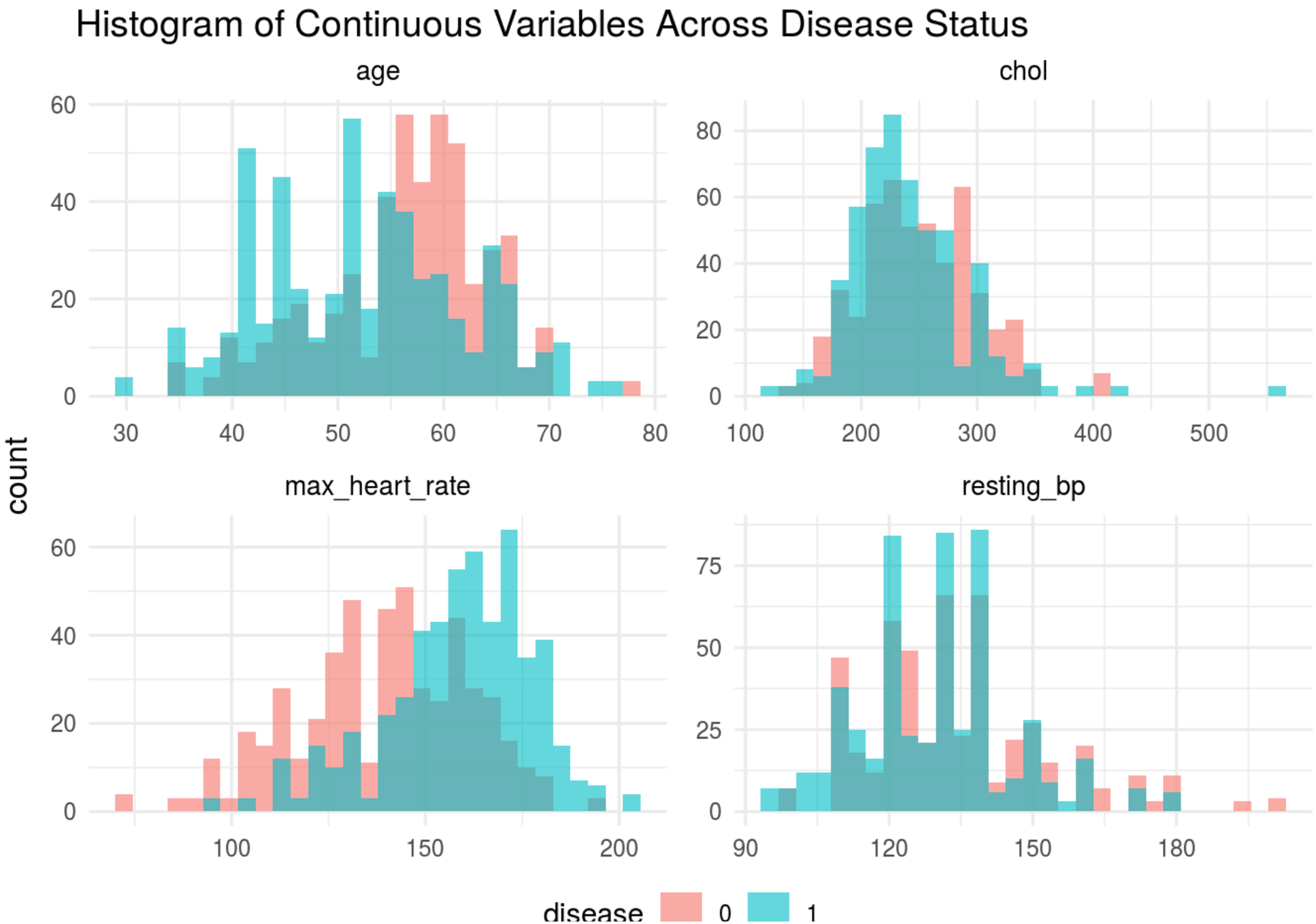



```
### First, create each plot separately and save as R objects

#predictor = age
age_plot <- ggplot(data = cvd_df_wide) +
  geom_histogram(aes(x=age, fill=disease), position = "identit
y",
                 alpha=0.6) +
  theme_minimal()+
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size=10)) +
```

```
  labs(title="age", x=NULL, y=NULL)

#predictor = cholesterol
chol_plot <- ggplot(data = cvd_df_wide) +
  geom_histogram(aes(x=chol, fill=disease), position = "identit
y",
                 alpha=0.6) +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size=10)) +
  labs(title="chol", x=NULL, y=NULL)

#predictor = maximum heart rate
max_heart_rate_plot <- ggplot(data = cvd_df_wide) +
  geom_histogram(aes(x=max_heart_rate, fill=disease),
                 position = "identity", alpha=0.6) +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size=10)) +
```

```
  labs(title="max_heart_rate", x=NULL, y=NULL)

#predictor = resting blood pressure
resting_bp_plot <- ggplot(data = cvd_df_wide) +
  geom_histogram(aes(x=resting_bp, fill=disease),
                 position = "identity", alpha=0.6) +
  theme_minimal() +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, size=10)) +
  labs(title="resting_bp", x=NULL, y=NULL)

#To create the 2x2 grid of plots, use ggpubr package
install.packages("ggpubr")
library(ggpubr)

#Use ggarrange to create grid and save as R object
arranged_plot <- ggarrange(age_plot, chol_plot, max_heart_rate_p
lot,
          resting_bp_plot, ncol = 2, nrow = 2,
```

```
          legend = "bottom", common.legend = TRUE)

#Add common title and y-axis label
annotate_figure(arranged_plot,
  top = text_grob("Histogram of Continuous Variables Across Dise
ase Status", size = 14, x = 0.4),
                bottom = NULL, left = "count", right = NULL)
```


Histogram of Continuous Variables Across Disease Status
age
chol
max_heart_rate
resting_bp
count
disease
0
1

c)

(1) Show the overall distribution of heart disease in the total dataset.

(2) Remove the id variable from the dataset, it may have been read in as `X1` *.*

(3) Before you split the data,set the seed to 5 in order to get reproducible results i.e run the line: `set.seed(5)` *.*

(4) Then split the data into a training and test set using a 75% split.

(5) Make sure the distribution of those positive for heart disease is similar in your training and test sets – you can do this with a visualization or a table.

```
#Show the overall distribution of heart disease in the total dat
aset
cvd_dist <- cvd_df_wide %>%
  group_by(disease) %>%
  summarise(freq = n()) %>%
  mutate(pct = freq / sum(freq))

knitr::kable(cvd_dist)
```

disease	freq	pct
0	499	0.4868293
1	526	0.5131707

```
#Remove ID variable
cvd_df_wide <- cvd_df_wide %>%
  select(-X1)

# Set a seed so that your results will not differ every time you
run the document
set.seed(5)

# Then split the data into a training and test set using a 75% s
plit
split_cvd_data <- initial_split(cvd_df_wide, prop = 0.75,
                                strata = disease)

cvd_train_dat <- training(split_cvd_data)

cvd_test_dat <- testing(split_cvd_data)

#Make sure the distribution of those positive for heart disease
is similar in your training and test sets
```

```
cvd_train_dist <- cvd_train_dat %>%
  group_by(disease) %>%
  summarise(freq = n()) %>%
  mutate(pct = freq / sum(freq))

cvd_test_dist <- cvd_test_dat %>%
  group_by(disease) %>%
  summarise(freq = n()) %>%
  mutate(pct = freq / sum(freq))

knitr::kable(cvd_train_dist)
```

disease	freq	pct
0	375	0.487013
1	395	0.512987

```
knitr::kable(cvd_test_dist)
```

disease	freq	pct
0	124	0.4862745
1	131	0.5137255

d)

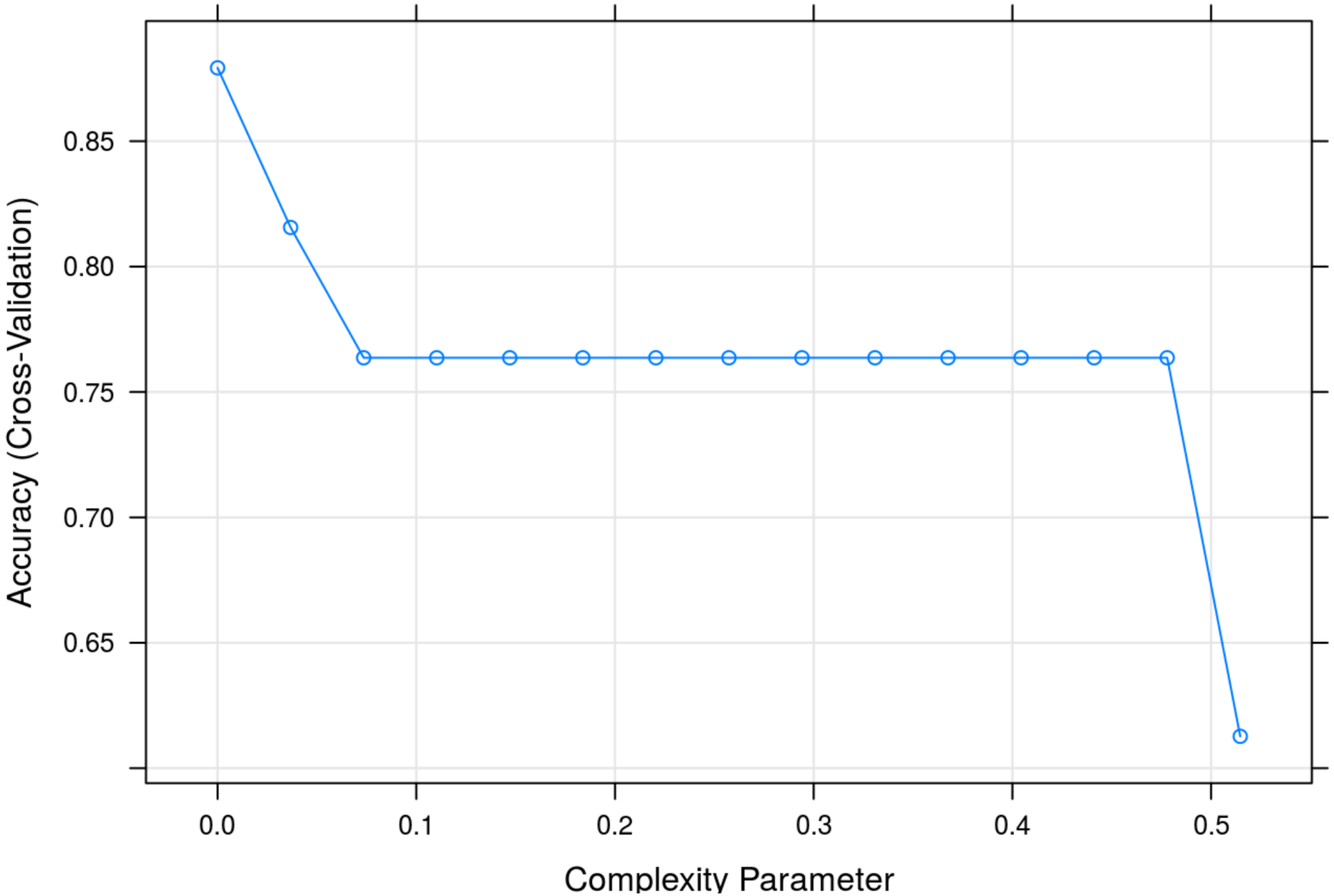
Train a classification tree (using `caret`) to classify heart disease status using all the variables in the data using 10-fold cross-validation. Have the cross-validation process consider 15 different `cp` values.

```
## Predict high bmi status using any variable (remember we removed BMI, height, and weight)

## Note that for classification problems, caret requires your outcome to be a two-level factor!

cvd_cart_cv <- train(
  form = disease ~ ., # formula
  data = cvd_train_dat, # training data
  method = "rpart", # CART
  trControl = trainControl(method = "cv",
                           number = 10), # 10-fold CV
  tuneLength = 15 # cross-validate over 15 different cp values
)
```

e)

Plot the model accuracy vs different complexity parameters and report which value maximizes the model accuracy (and report in a sentence what the best accuracy was).

```
## You can immediately plot the results of your cross-validation using plot():
plot(cvd_cart_cv)
```


Accuracy (Cross-Validation)
0.85
0.80
0.75
0.70
0.65
0.0
0.1
0.2
0.3
0.4
0.5
Complexity Parameter

```
## caret provides the best cross-validated cp under the name "be
stTune"

cvd_cart_cv$bestTune
```

```
##   cp
## 1  0
```

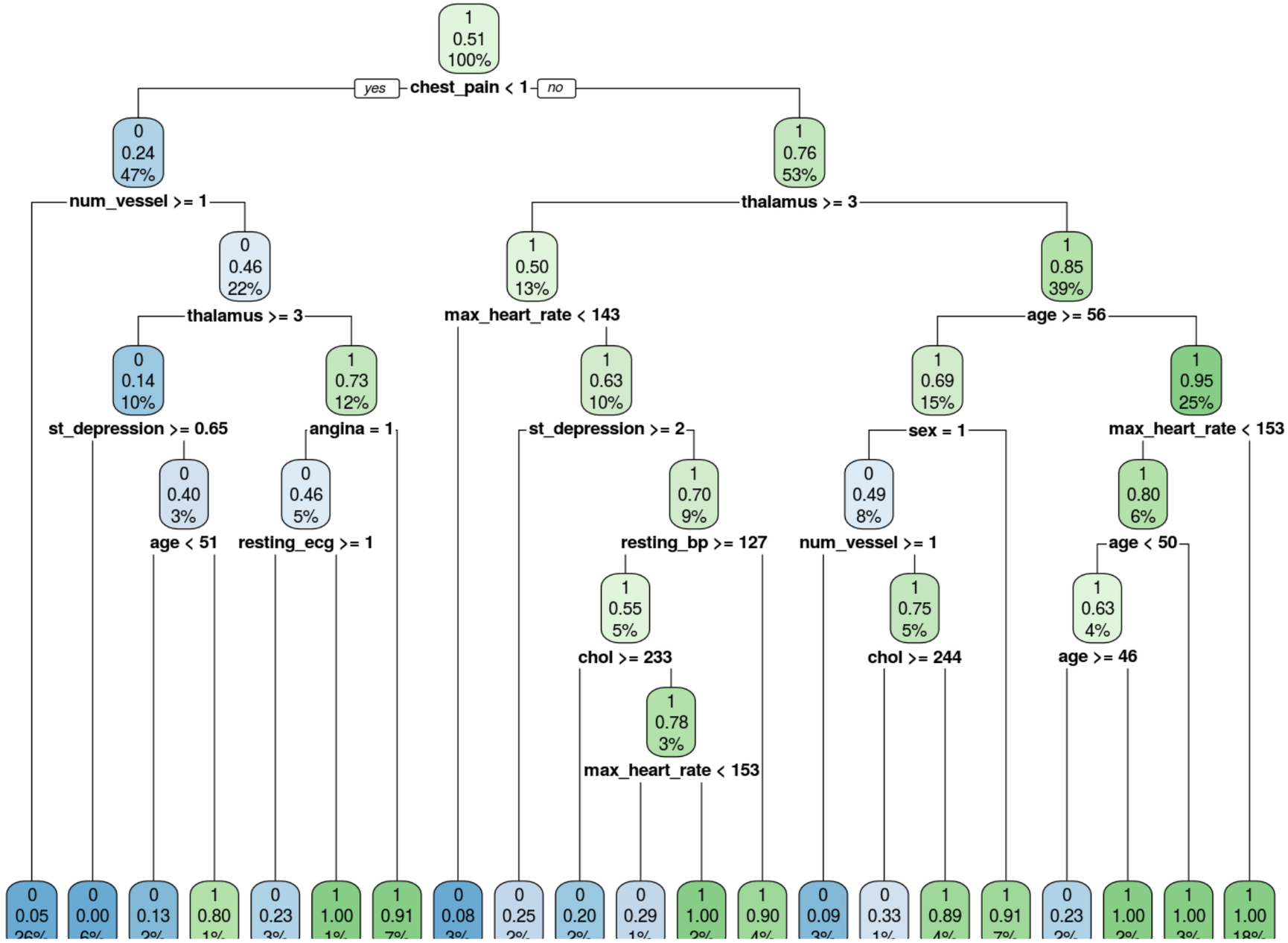
```
## you can take a closer look at the results for every cp value:

cvd_cart_cv_details <- cvd_cart_cv$results

#Find cp for which accuracy is highest
cvd_cp_best <- cvd_cart_cv_details %>%
  arrange(-Accuracy) %>%
  slice(1)
```

The complexity parameter that maximizes model accuracy is 0, and the accuracy is 0.8792291.

f)

Create a plot/diagram of the best (final) classification tree model from the cross-validation process on the training data using `rpart.plot`*. In order to see the plot properly you may have to knit your HTML, we have included* `fig.width` *and* `fig.height` *options in the code chunk below to create a big enough plot to read.*

```
## caret will also provide you with the model trained to the tra
ining data using the best cross-validated cp:

cvd_final_model <- cvd_cart_cv$finalModel

## The rpart.plot package provides nice plots for classification
and regression trees:
## more info available here: http://www.milbo.org/rpart-plot/pr
p.pdf

rpart.plot(cvd_cart_cv$finalModel)
```


1
0.51
100%
yes
chest_pain < 1
no
0
0.24
47%
1
0.76
53%
num_vessel >= 1
thalamus >= 3
0
0.46
22%
1
0.50
13%
1
0.85
39%
thalamus >= 3
max_heart_rate < 143
age >= 56
0
0.14
10%
1
0.73
12%
1
0.63
10%
1
0.69
15%
1
0.95
25%
st_depression >= 0.65
angina = 1
st_depression >= 2
sex = 1
max_heart_rate < 153
0
0.40
3%
0
0.46
5%
1
0.70
9%
0
0.49
8%
1
0.80
6%
age < 51
resting_ecg >= 1
resting_bp >= 127
num_vessel >= 1
age < 50
1
0.55
5%
1
0.75
5%
1
0.63
4%
chol >= 233
chol >= 244
age >= 46
1
0.78
3%
max_heart_rate < 153
0 0.05 26%
0 0.00 6%
0 0.13 2%
1 0.80 1%
0 0.23 3%
1 1.00 1%
1 0.91 7%
0 0.08 3%
0 0.25 2%
0 0.20 2%
0 0.29 1%
1 1.00 2%
1 0.90 4%
0 0.09 3%
0 0.33 1%
1 0.89 4%
1 0.91 7%
0 0.23 2%
1 1.00 2%
1 1.00 3%
1 1.00 18%

g)

Take a look at your classification tree diagram from (f) (it may be easier to look at in a knit html. What is the root node of the tree and what is the splitting rule at the root node? How many leaf nodes are there in this tree?.

The root node is chest pain, and the root node splits on whether or not the patient has chest pain (yes or no). There are 21 leaf nodes in this tree.

h)

Using your classification tree diagram from (f), trace the path of a person who is 60 years old, has a chest pain score of 0, is positive for angina, has a thalamus score of 3, a num_vessel score of 2, and a resting ecg score of 1. Will your classification tree predict that this person has heart disease or not? Which leaf node does this person end up in? (you can count starting from the left side of the graph).

Yes, the classification tree predicts this person has heart disease. This person ends up in the 7th leaf node from the left.

i)

Train a logistic regression model (using `caret`) to classify heart disease status using all the variables in the data. Report the cross-validated accuracy (10-fold) of this model on the training dataset.

```
cvd_logistic_cv <- train(
  form = disease ~ ., # include everything as predictors
  data = cvd_train_dat,
  trControl = trainControl(method = "cv",
                           number = 10), # 10-fold cv
  method = "glm",
  family = "binomial"
)

cvd_logit_df <- cvd_logistic_cv$results

summary(cvd_logistic_cv)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5143  -0.3916   0.1114   0.5896   2.6782
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.790629   1.577353   2.403 0.016254 *
## age           -0.002286   0.014480  -0.158 0.874533
## sex           -1.708774   0.291864  -5.855 4.78e-09 ***
## chest_pain     0.887446   0.116905   7.591 3.17e-14 ***
## resting_bp    -0.022490   0.006523  -3.448 0.000565 ***
## chol          -0.005062   0.002466  -2.052 0.040127 *
## fbs           -0.145943   0.328038  -0.445 0.656395
## resting_ecg    0.220168   0.215155   1.023 0.306164
## max_heart_rate  0.023573   0.006452   3.654 0.000259 ***
```

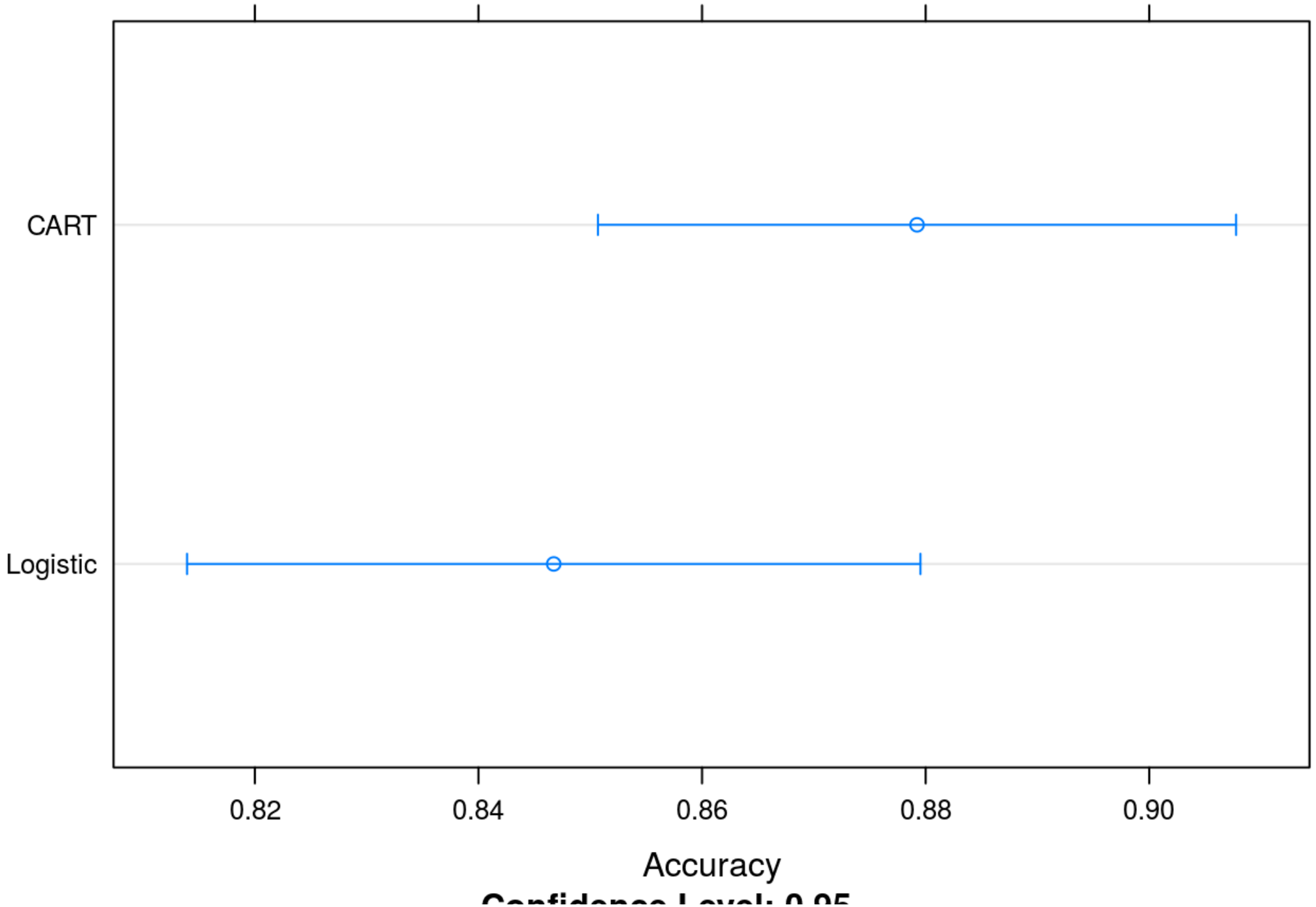
```
## angina          -0.904193   0.259874  -3.479 0.000503 ***
## st_depression   -0.545458   0.132250  -4.124 3.72e-05 ***
## slope            0.523978   0.214659   2.441 0.014647 *
## num_vessel      -0.753758   0.119718  -6.296 3.05e-10 ***
## thalamus        -0.912013   0.179640  -5.077 3.84e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1066.93  on 769  degrees of freedom
## Residual deviance:  543.64  on 756  degrees of freedom
## AIC: 571.64
##
## Number of Fisher Scoring iterations: 6
```

The accuracy of the logistic regression model was 0.8467331.

j)

Compare the cross-validated accuracy of the classification tree and logistic regression model. You can do this with the `resamples()` and the `dotpot()` functions as we did in Lab 11. Which model performs better on the training data?

```
# Use the "resamples" function with a list of all your caret-tra
ined objects:

comparison <- resamples(list("Logistic"=cvd_logistic_cv,
                             "CART" = cvd_cart_cv))



dotplot(comparison, metric = "Accuracy")
```


CART
Logistic
0.82
0.84
0.86
0.88
0.90
Accuracy

Confidence Level: 0.95

The point estimate of accuracy for the classification tree was higher, but the 95% confidence intervals of accuracy for both models overlap substantially, suggesting neither model was significantly better than the other.

k)

Choose whichever model you think has the best cross-validated performance in the training data as your final model. Then calculate your final model's accuracy on the test set. What do you think – would you use this model to predict the presence of heart disease in future patients?

```
## CHOOSING THE CART MODEL:

# caret has built-in functionality to make predictions on the te
st dataset

cvd_cart_predictions <- predict(cvd_cart_cv, cvd_test_dat)

cvd_cart_pred_dat <- cvd_test_dat %>%
  mutate(cvd_predictions = predict(cvd_cart_cv, cvd_test_dat))



# To obtain accuracy (and other) measures of predictive performa
nce we can use the confusionMatrix function from caret:

cvd_cart_performance_final <- confusionMatrix(
  data = cvd_cart_pred_dat$cvd_predictions,
  reference = cvd_cart_pred_dat$disease)
```

```
cvd_cart_performance_final
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 117  13
##          1   7 118
##
##                Accuracy : 0.9216
##                  95% CI : (0.8815, 0.9514)
##     No Information Rate : 0.5137
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8432
##
##  Mcnemar's Test P-Value : 0.2636
##
##             Sensitivity : 0.9435
##             Specificity : 0.9008
##          Pos Pred Value : 0.9000
```

```
##          Neg Pred Value : 0.9440
##              Prevalence : 0.4863
##          Detection Rate : 0.4588
##    Detection Prevalence : 0.5098
##       Balanced Accuracy : 0.9222
##
##        'Positive' Class : 0
##
```

```
cvd_cart_performance_final <- cvd_cart_performance_final %>%
  tidy()

# We can also create our own confusion matrix to check:

cvd_cart_true_pos_n <- cvd_cart_pred_dat %>%
  filter(disease == 1 & cvd_predictions == 1) %>%
  nrow()

cvd_cart_true_neg_n <- cvd_cart_pred_dat %>%
  filter(disease == 0 & cvd_predictions == 0) %>%
  nrow()

cvd_cart_accuracy <- (cvd_cart_true_pos_n + cvd_cart_true_neg_
n)/nrow(cvd_cart_pred_dat)

cvd_cart_accuracy
```

```
## [1] 0.9215686
```

The final model's accuracy is 0.9215686 with a 95% confidence interval of (0.881467, 0.9514354).

Yes, I would use this model to predict CVD in future patients because it has high accuracy and high specificity (0.9435484). To predict CVD, I'd favor a model with high specificity because there are deleterious effects of CVD that need to be addressed if a patient has it. Lifestyle changes recommended to address CVD (e.g. healthier diet, more exercise) could still benefit someone who doesn't have CVD but tests positive for it according to the model.

Question 2

The last question this semester is quite different: the purpose of this question is to go back and try to summarize the takeaway of each lecture. We know that this semester feels like it has lasted a century so we thought that this type of question

will help you to synthesize how far we have come and how much we have learned together.

Please look back at each lecture and summarize in words (you can use formulas or images too) what the biggest take away of each lecture was for you:

Lecture 1 (01/21)

Introduction to R tools and to the course.

Lecture 2 (01/28)

Introduction to data frames and ggplot.

Lecture 3 (02/04)

Review of ggplot, introduction to basic dplyr functions to manipulate data frames, and review of hypothesis testing.

Lecture 4 (02/11)

Learning more aspects of ggplot that enhance graph-making/data visualization, review of error types and power calculation.

Lecture 5 (02/18)

Computing power in R, creating functions, and extracting items from vectors & lists.

Lecture 6 (02/25)

Review of functions, extracting from vectors & lists, and of Central Limit Theorem. Also, learning how to merge data frames.

Lecture 7 (03/03)

Permutation tests and linear regression (statistical theory and R coding). Also, learning how to organize regression results into a data frame.

Lecture 8 (03/31)

Review of linear regression (statistical theory and R coding) as well as introduction to string manipulation.

Lecture 9 (04/07)

Review of string manipulation and permutation tests (statistical theory and R coding) as well as introduction to bootstrapping (statitical theory and R coding) and machine learning (theory).

Lecture 10 (04/14)

Practical uses of bootstrapping and machine learning.

Lecture 11 (04/21)

Machine learning application and introduction to logistic regression and decision trees (statistical theory and R coding).