# Final Project

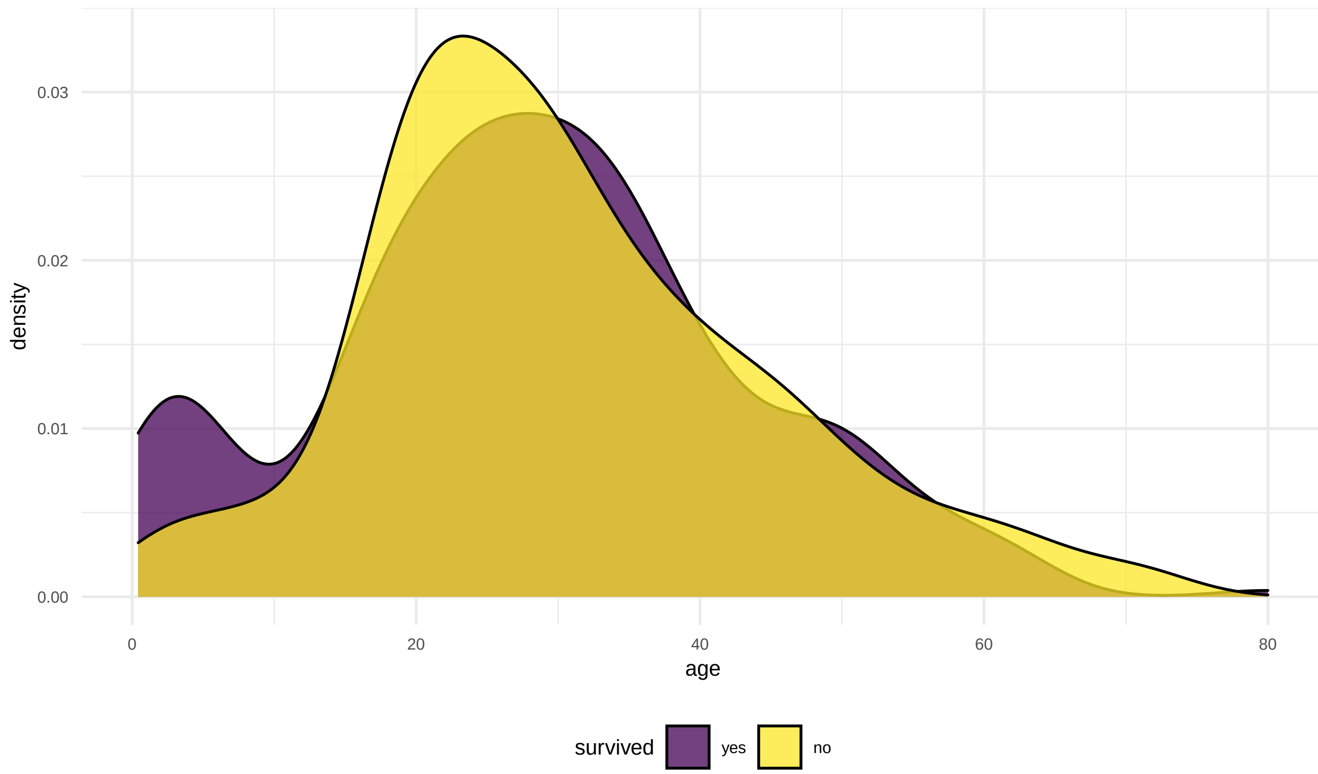
David Nemirovsky & Jared Klug

5/13/21

## EDA

```
titanic_df =
  read_csv("./data/train.csv") %>%
  janitor::clean_names() %>%
  mutate(survived = fct_recode(as.factor(survived), yes = "1", no = "0"),
         survived = fct_relevel(survived, "yes", "no"),
         pclass = as.factor(pclass),
         sex = as.factor(sex),
         embarked = as.factor(embarked)) %>%
  select(-c(ticket, cabin, name, passenger_id))

# Missing Data EDA:
vis_miss(titanic_df)
```

```r
titanic_df %>%
  ggplot(aes(x = age, fill = survived)) +
  geom_density(alpha = 0.75)
```




```r
# Missing Survivals:
titanic_df %>%
  filter(is.na(age)) %>%
  group_by(survived) %>%
  summarise(count = n())
```

```
## # A tibble: 2 x 2
##   survived count
##   <fct>    <int>
## 1 yes         52
## 2 no         125
```

```r
# All survivals:
titanic_df %>%
  group_by(survived) %>%
  summarise(count = n())
```

```
## # A tibble: 2 x 2
##   survived count
##   <fct>    <int>
## 1 yes        342
## 2 no         549
```

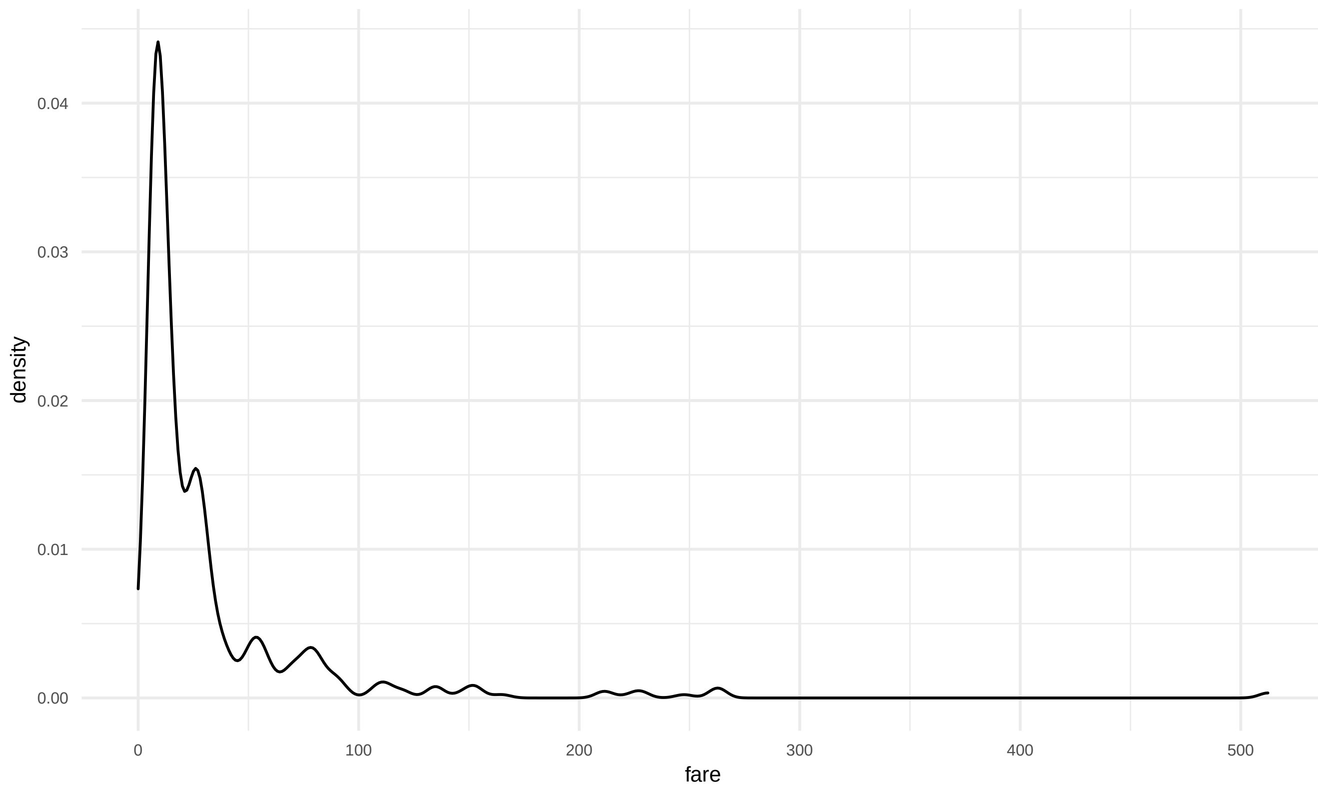
```r
# Survival percentage in NAs is similar to actual survival rate so assume missingness is not related to

# Bagging imputation will be used to fill in missing age data, embarked will be replaced by the most po

titanic_df %>%
  group_by(embarked) %>%
  summarise(count = n())
```

```
## # A tibble: 4 x 2
##   embarked count
##   <fct>    <int>
## 1 C          168
## 2 Q           77
## 3 S          644
## 4 <NA>         2
```

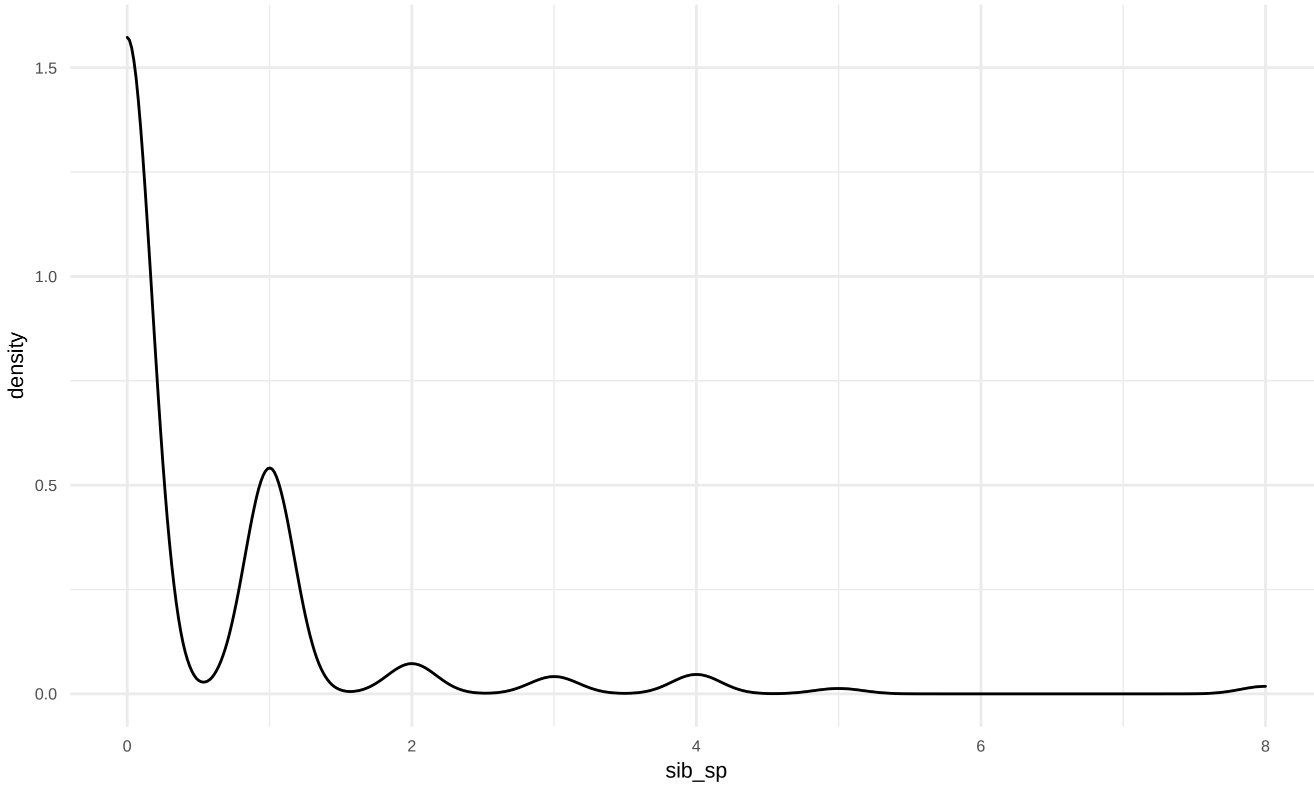
```r
titanic_df[is.na(titanic_df$embarked),"embarked"] = "S"

trainX = titanic_df[-1]
train_bag = preProcess(trainX, method = "bagImpute")
train_imp = predict(train_bag, trainX)
train_df = cbind(titanic_df[1], train_imp)

# Descriptive Summary Table:
eda_df =
  train_df %>%
  mutate(survived = fct_recode(as.factor(survived),
                          Survived = "yes", Died = "no"),
         survived = fct_relevel(survived, "Died", "Survived"),
         sex = fct_recode(as.factor(sex),
                          Female = "female", Male = "male"),
         embarked = fct_recode(as.factor(embarked),
                               Cherbourg = "C",
                               Queenstown = "Q",
                               Southampton = "S"),
         pclass = fct_recode(as.factor(pclass),
                             Upper = "1",
                             Middle = "2",
                             Lower = "3"))
eda_df %>%
  tbl_summary(by = survived,
              label =
                list(
                  pclass ~ "Socioeconomic Status",
                  sex ~ "Sex",
                  age ~ "Age",
                  sib_sp ~ "Number of Siblings/Spouse on Board",
                  parch ~ "Number of Parents/Children on Board",
                  fare ~ "Passenger Fare",
                  embarked ~ "Port of Embarkation")) %>%
  add_overall
```

| **Characteristic** | **Overall**, N = 891 | **Died**, N = 549 | **Survived**, N = 342 |
|---|---|---|---|
| Socioeconomic Status | | | |
| Upper | 216 (24%) | 80 (15%) | 136 (40%) |
| Middle | 184 (21%) | 97 (18%) | 87 (25%) |
| Lower | 491 (55%) | 372 (68%) | 119 (35%) |
| Sex | | | |
| Female | 314 (35%) | 81 (15%) | 233 (68%) |
| Male | 577 (65%) | 468 (85%) | 109 (32%) |
| Age | 28 (22, 36) | 29 (22, 36) | 28 (20, 36) |
| Number of Siblings/Spouse on Board | | | |
| 0 | 608 (68%) | 398 (72%) | 210 (61%) |
| 1 | 209 (23%) | 97 (18%) | 112 (33%) |
| 2 | 28 (3.1%) | 15 (2.7%) | 13 (3.8%) |
| 3 | 16 (1.8%) | 12 (2.2%) | 4 (1.2%) |
| 4 | 18 (2.0%) | 15 (2.7%) | 3 (0.9%) |
| 5 | 5 (0.6%) | 5 (0.9%) | 0 (0%) |
| 8 | 7 (0.8%) | 7 (1.3%) | 0 (0%) |
| Number of Parents/Children on Board | | | |
| 0 | 678 (76%) | 445 (81%) | 233 (68%) |
| 1 | 118 (13%) | 53 (9.7%) | 65 (19%) |
| 2 | 80 (9.0%) | 40 (7.3%) | 40 (12%) |
| 3 | 5 (0.6%) | 2 (0.4%) | 3 (0.9%) |
| 4 | 4 (0.4%) | 4 (0.7%) | 0 (0%) |
| 5 | 5 (0.6%) | 4 (0.7%) | 1 (0.3%) |
| 6 | 1 (0.1%) | 1 (0.2%) | 0 (0%) |
| Passenger Fare | 14 (8, 31) | 10 (8, 26) | 26 (12, 57) |
| Port of Embarkation | | | |
| Cherbourg | 168 (19%) | 75 (14%) | 93 (27%) |
| Queenstown | 77 (8.6%) | 47 (8.6%) | 30 (8.8%) |
| Southampton | 646 (73%) | 427 (78%) | 219 (64%) |

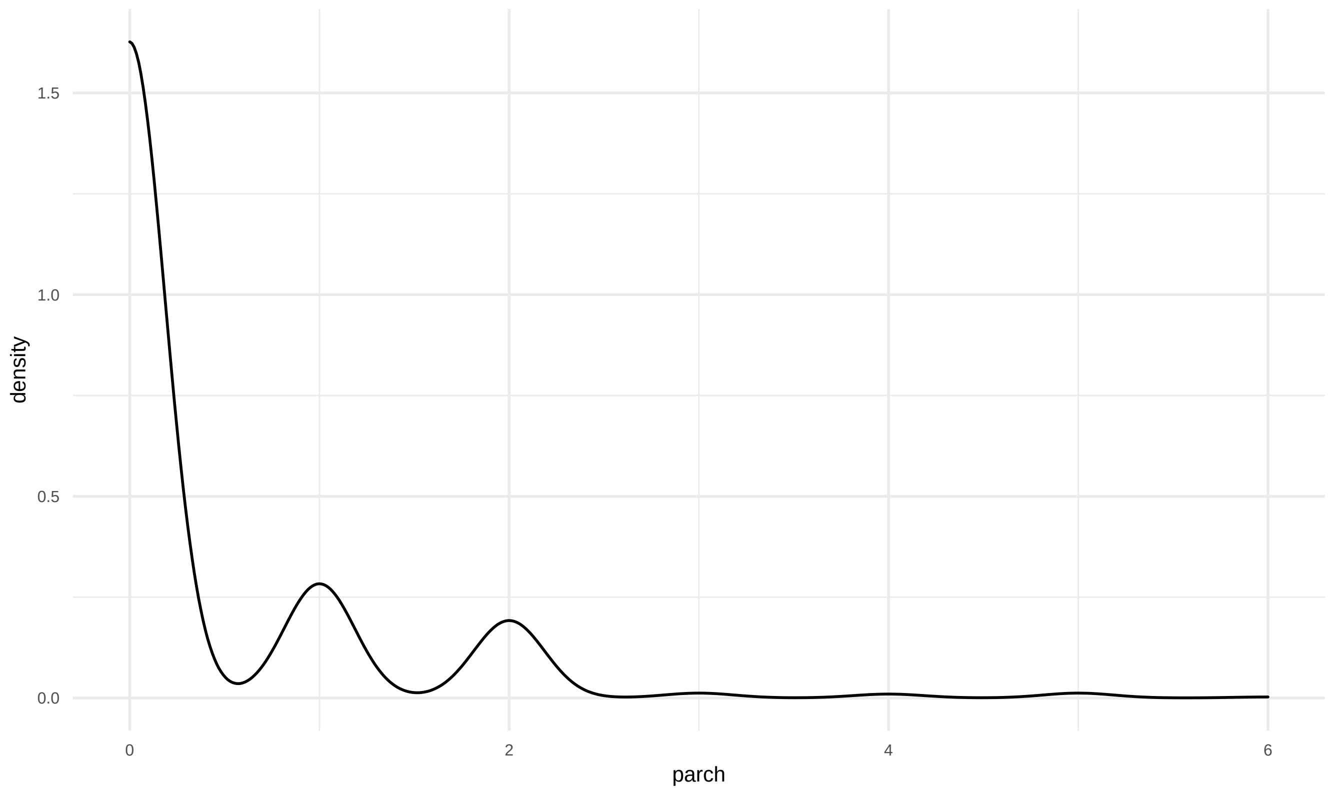
```r
#featurePlot(x = select(mutate(train_df,
#                              pclass = as.numeric(pclass),
#                              sex = as.numeric(sex),
#                              embarked = as.numeric(embarked)),
#                       pclass:embarked),
#            y = train_df$survived,
#            scales = list(x = list(relation = "free"),
#                          y = list(relation = "free")),
#            plot = "density",
#            auto.key = list(columns = 2))

eda_df %>%
  ggplot(aes(x = fare)) +
  geom_density()
```




```r
eda_df %>%
  ggplot(aes(x = sib_sp)) +
  geom_density()
```

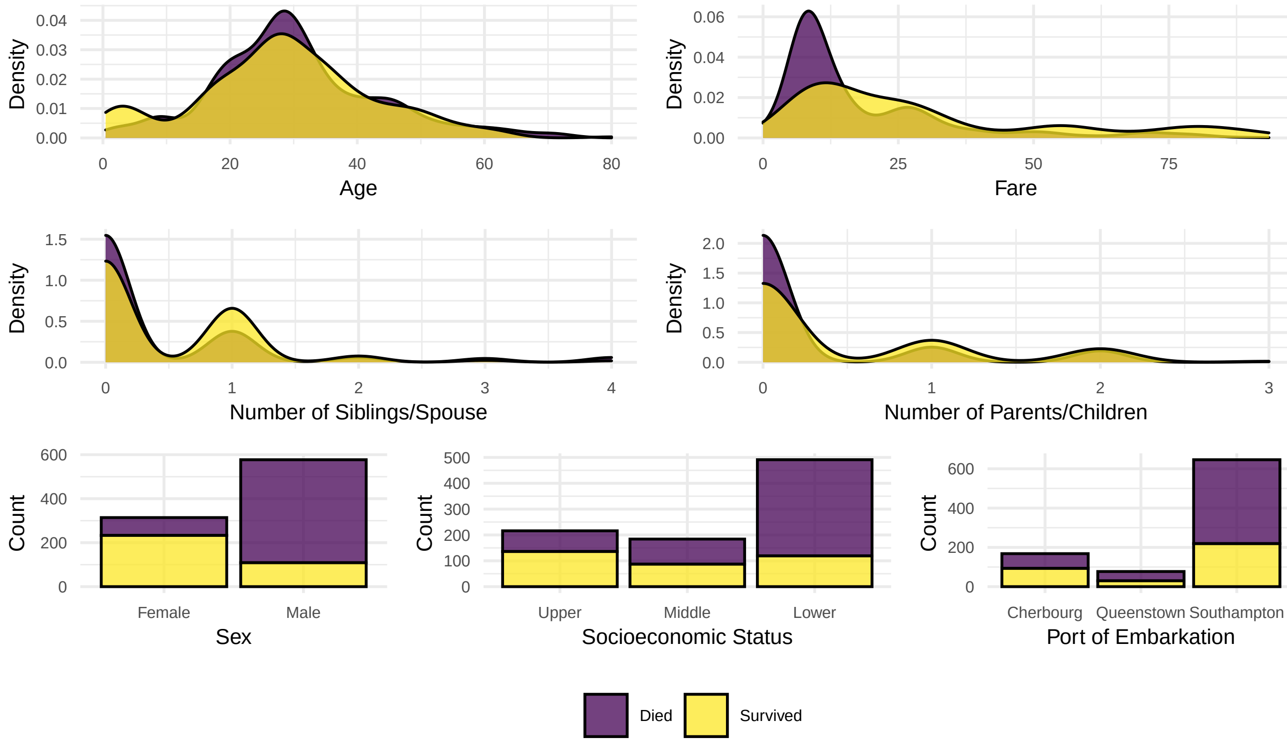



```r
eda_df %>%
  ggplot(aes(x = parch)) +
  geom_density()
```

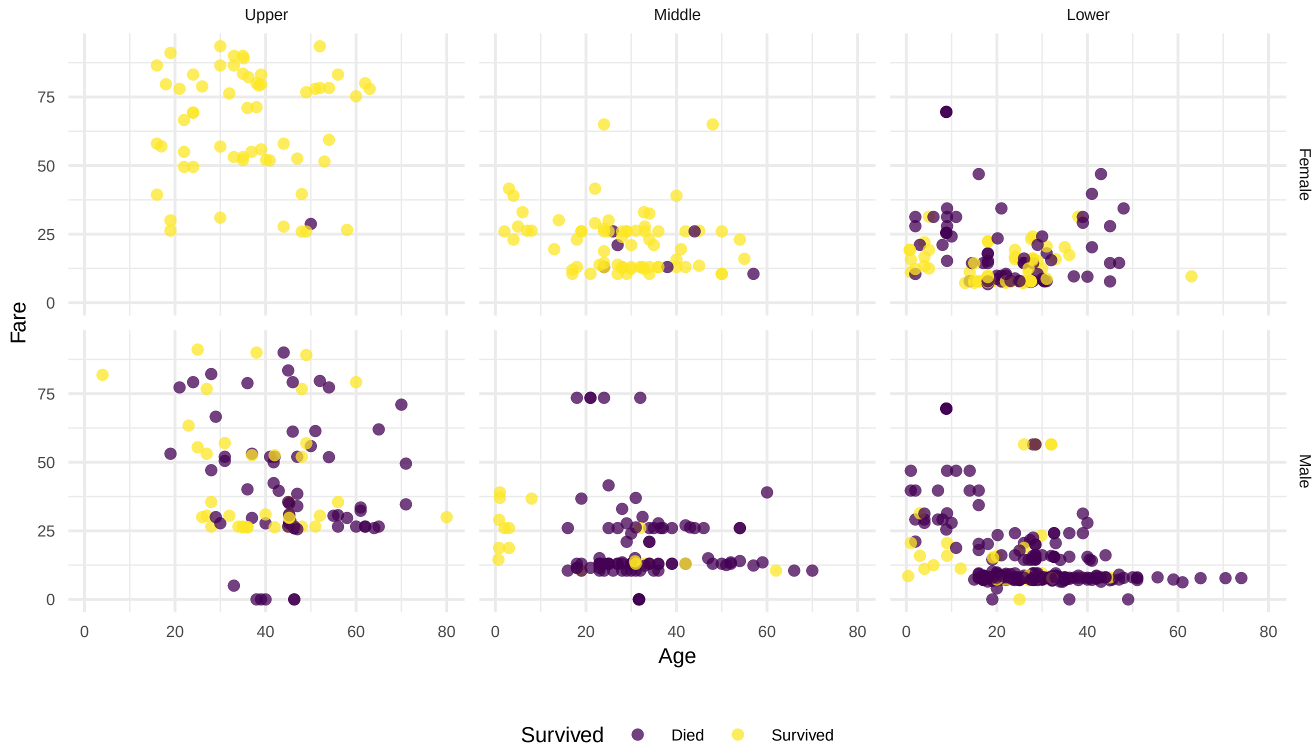



```r
# Survival by Age:
plot_age =
  eda_df %>%
  ggplot(aes(x = age, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Age", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Fare:
plot_fare =
  eda_df %>%
  filter(fare < 100) %>%
  ggplot(aes(x = fare, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Fare", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Number of Siblings/Spouse:
plot_sibsp =
  eda_df %>%
  filter(sib_sp < 5) %>%
  ggplot(aes(x = sib_sp, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Number of Siblings/Spouse", y = "Density") +
  scale_fill_discrete(name = "Survived") +
```

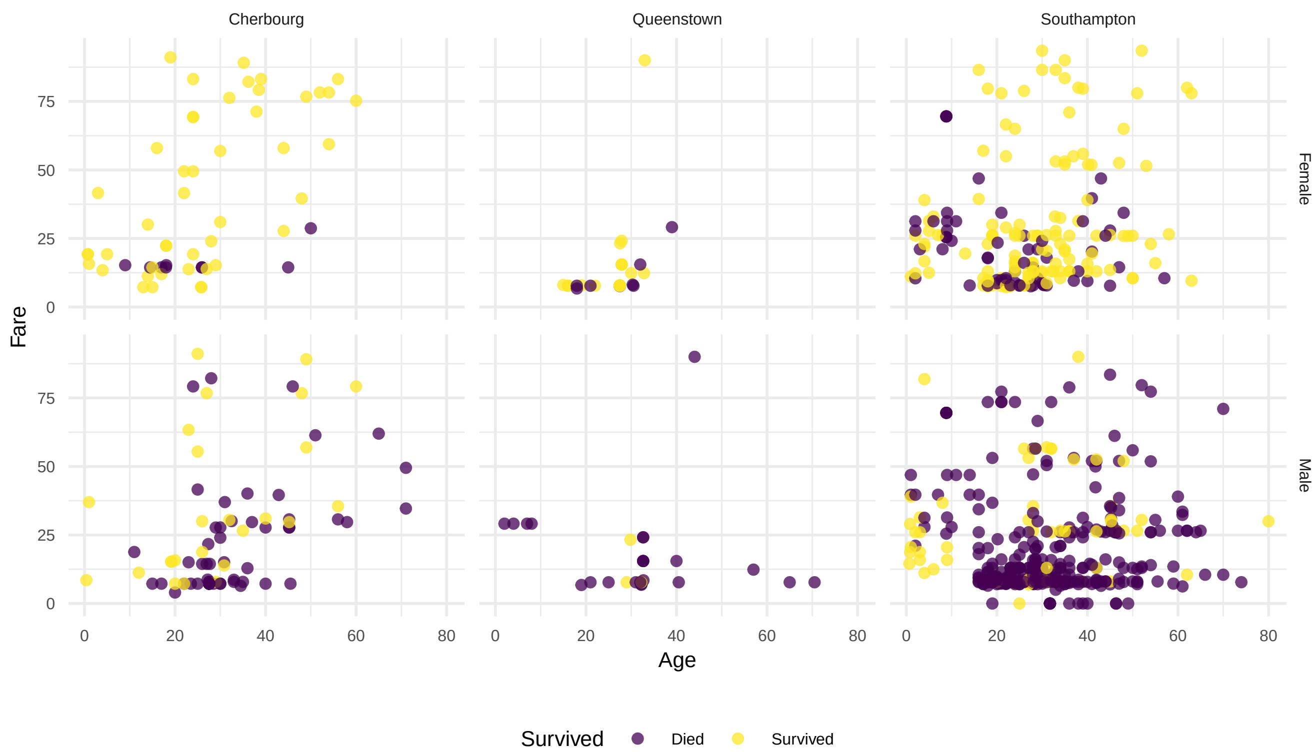
```
  theme(legend.title = element_blank())

# Survival by Number of Parents/Children:
plot_parch =
  eda_df %>%
  filter(parch < 4) %>%
  ggplot(aes(x = parch, fill = survived)) +
  geom_density(alpha = 0.75) +
  labs(x = "Number of Parents/Children", y = "Density") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Sex:
plot_sex =
  eda_df %>%
  ggplot(aes(x = sex, fill = survived)) +
  geom_bar(color = "black", alpha = 0.75) +
  labs(x = "Sex", y = "Count") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Socioeconomic Status:
plot_pclass =
  eda_df %>%
  ggplot(aes(x = pclass, fill = survived)) +
  geom_bar(color = "black", alpha = 0.75) +
  labs(x = "Socioeconomic Status", y = "Count") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

# Survival by Port of Embarkation:
plot_emb =
  eda_df %>%
  ggplot(aes(x = embarked, fill = survived)) +
  geom_bar(color = "black", alpha = 0.75) +
  labs(x = "Port of Embarkation", y = "Count") +
  scale_fill_discrete(name = "Survived") +
  theme(legend.title = element_blank())

layout = "
AAABBB
CCCDDD
EEFFGG
"
plot_age + plot_fare + plot_sibsp + plot_parch + plot_sex +
  plot_pclass + plot_emb + plot_layout(design = layout, guides = "collect")
```




```r
# Survival of Age vs Fare by SES and Sex
eda_df %>%
  filter(fare < 100) %>%
  ggplot(aes(x = age, y = fare, color = survived)) +
  geom_point(alpha = 0.75) +
  facet_grid(sex ~ pclass) +
  labs(x = "Age", y = "Fare") +
  scale_color_discrete(name = "Survived")
```




```r
# Survival of Age vs Fare by Embarkation and Sex
eda_df %>%
  filter(fare < 100) %>%
  ggplot(aes(x = age, y = fare, color = survived)) +
  geom_point(alpha = 0.75) +
  facet_grid(sex ~ embarked) +
  labs(x = "Age", y = "Fare") +
  scale_color_discrete(name = "Survived")
```




## Model Training

```
ctrl = trainControl(method = "repeatedcv", summaryFunction = twoClassSummary, classProbs = T, number =

set.seed(37564)
mod_enet = train(survived ~ .,
                 na.action = na.exclude,
                 data = train_df,
                 method = "glmnet",
                 family = "binomial",
                 metric = "ROC",
                 tuneGrid = expand.grid(alpha = seq(0, 0.5, length = 6),
                                        lambda = exp(seq(-4, -8, length = 50))),
                 trControl = ctrl)
tuning_plot_enet =
  ggplot(mod_enet, highlight = T) +
  ggtitle("Elastic Net") +
  theme(plot.title = element_text(hjust = 0.5))
mod_enet$bestTune

set.seed(37564)
mod_mars = train(survived ~ .,
                 na.action = na.exclude,
                 data = train_df,
                 method = "earth",
                 tuneGrid = expand.grid(degree = 1:3, nprune = 5:15),
```

```r
                  metric = "ROC",
                  trControl = ctrl)
tuning_plot_mars =
  ggplot(mod_mars, highlight = T) +
  ggtitle("MARS") +
  theme(plot.title = element_text(hjust = 0.5))
mod_mars$bestTune

set.seed(37564)
mod_knn = train(survived ~ .,
                na.action = na.exclude,
                data = train_df,
                method = "knn",
                metric = "ROC",
                preProcess = c("center","scale"),
                tuneGrid = data.frame(k = seq(1, 30, by = 1)),
                trControl = ctrl)
tuning_plot_knn =
  ggplot(mod_knn, highlight = T) +
  ggtitle("KNN") +
  theme(plot.title = element_text(hjust = 0.5))
mod_knn$bestTune

set.seed(37564)
mod_boost = train(survived ~ .,
                  na.action = na.exclude,
                  data = train_df,
                  method = "gbm",
                  distribution = "adaboost",
                  tuneGrid = expand.grid(n.trees = c(2000, 3000),
                                         interaction.depth = 4:13,
                                         shrinkage = c(0.003, 0.005, 0.007),
                                         n.minobsinnode = 1),
                  metric = "ROC",
                  trControl = ctrl,
                  verbose = F)
tuning_plot_boost =
  ggplot(mod_boost, highlight = T) +
  ggtitle("Boosting") +
  theme(plot.title = element_text(hjust = 0.5))
mod_boost$bestTune

set.seed(37564)
mod_svm = train(survived ~ .,
                na.action = na.exclude,
                data = train_df,
                preProcess = c("scale", "center"),
                method = "svmRadialSigma",
                tuneGrid = expand.grid(C = exp(seq(-2,3, len = 10)),
                                       sigma = exp(seq(-8,0, len = 10))),
                metric = "ROC",
                trControl = ctrl)
tuning_plot_svm =
```

```r
  ggplot(mod_svm, highlight = T) +
  ggtitle("SVM Radial") +
  theme(plot.title = element_text(hjust = 0.5))
mod_svm$bestTune

layout2 = "
AABB
CCDD
EEEE
"
tuning_plot_knn + tuning_plot_enet +
  tuning_plot_mars + tuning_plot_svm +
  tuning_plot_boost + plot_layout(design = layout2)

res = resamples(list(ENET = mod_enet, MARS = mod_mars, KNN = mod_knn, BOOST = mod_boost, SVM = mod_svm))
summary(res)
bwplot(res, metric = "ROC", main = "ROC for Repeated 10-Fold CV Using Various Models")
```

### Variable Importance

```r
set.seed(37564)

vip(mod_boost,
    method = "permute",
    train = train_df,
    target = "survived",
    metric = "auc",
    reference_class = c("yes", "no"),
    nsim = 30,
    pred_wrapper = predict,
    geom = "boxplot",
    all_permutations = T,
    mapping = aes_string(fill = "Variable", alpha = 0.75))

#Check if enet parameters micking importance pattern
coef(mod_enet$finalModel, mod_enet$bestTune$lambda)
```

### Predictions

```r
testna_df =
  read_csv("./data/test.csv") %>%
  janitor::clean_names() %>%
  select(-c(ticket, cabin, name)) %>%
  left_join(janitor::clean_names(read_csv("./data/titanic_results.csv"))) %>%
  mutate(pclass = as.factor(pclass),
         sex = as.factor(sex),
         embarked = as.factor(embarked))

testX = testna_df[,2:8]
```

```r
test_bag = preProcess(testX, method = "bagImpute")
test_df = predict(test_bag, testX) %>%
  cbind(testna_df[1], testna_df[9])

pred_knn = predict(mod_knn, newdata = test_df, type = "prob")[,1]
roc_knn = roc(test_df$survived, pred_knn)
plot(roc_knn, legacy.axes = T)

pred_enet = predict(mod_enet, newdata = test_df, type = "prob")[,1]
roc_enet = roc(test_df$survived, pred_enet)
plot(roc_enet, add = T, col = 2)

pred_svm = predict(mod_svm, newdata = test_df, type = "prob")[,1]
roc_svm = roc(test_df$survived, pred_svm)
plot(roc_svm, add = T, col = 3)

pred_mars = predict(mod_mars, newdata = test_df, type = "prob")[,1]
roc_mars = roc(test_df$survived, pred_mars)
plot(roc_mars, add = T, col = 4)

pred_boost = predict(mod_boost, newdata = test_df, type = "prob")[,1]
roc_boost = roc(test_df$survived, pred_boost)

auc = c(roc_knn$auc[1], roc_enet$auc[1], roc_svm$auc[1], roc_mars$auc[1], roc_boost$auc[1])
modelNames = c("KNN","Elastic Net", "SVM", "MARS", "Boosting")
legend("bottomright", legend = paste0(modelNames, ": ", round(auc,3)),
       col = c(1:4, 7), lwd = 2)
plot(roc_boost, add = T, col = 7)
```

### Boosted RF Model Analysis

```r
cm_df = pred_boost %>%
  as.data.frame() %>%
  rename("survived" = ".") %>%
  mutate(survived = as.factor(ifelse(survived >= 0.5, 1, 0)))

confusionMatrix(data = cm_df$survived, reference = as.factor(test_df$survived))

explainer_gbm = explain(mod_boost,
                        label = "gbm",
                        data = train_df[-1],
                        y = as.numeric(train_df$survived == "yes"),
                        verbose = F)

vi_gbm = model_parts(explainer_gbm)

plot(vi_gbm)

pdp_age = model_profile(explainer_gbm,
                        variable = "age",
                        type = "partial")
```

```r
pdp_fare = model_profile(explainer_gbm,
                         variable = "fare",
                         type = "partial")

plot(pdp_age, pdp_fare)

pb_gbm = predict_parts(explainer_gbm,
                       new_observation = test_df[1,],
                       type = "break_down")

plot(pb_gbm)
```