```r
options(warn = -1)
library(tidyverse)
library(ggthemes)
library(ggplot2)
library(tidymodels)
library(corrplot)
library(yardstick)
library(ISLR)
library(ISLR2)
library(discrim)
library(corrr)
tidymodels_prefer()
```

```r
titanic <- read.csv(file="/Users/honchowayne/Desktop/titanic.csv")
titanic %>% head()
```

```
##   passenger_id survived pclass
## 1            1       No      3
## 2            2      Yes      1
## 3            3      Yes      3
## 4            4      Yes      1
## 5            5       No      3
## 6            6       No      3
##                                                  name    sex age sib_sp parch
## 1                             Braund, Mr. Owen Harris   male  22      1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38      1     0
## 3                              Heikkinen, Miss. Laina female  26      0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35      1     0
## 5                            Allen, Mr. William Henry   male  35      0     0
## 6                                    Moran, Mr. James   male  NA      0     0
##             ticket    fare cabin embarked
## 1        A/5 21171  7.2500  <NA>        S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250  <NA>        S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500  <NA>        S
## 6           330877  8.4583  <NA>        Q
```

```r
titanic$survived <- as.factor(titanic$survived)
titanic$survived <- factor(titanic$survived, levels = c("Yes" , "No"))
titanic$pclass <- as.factor(titanic$pclass)
titanic$sex <- as.factor(titanic$sex)
```

Question1: Split the data

```r
set.seed(3435)
titanic_split <- initial_split(titanic,strata = survived, prop = 0.80)
titanic_train <- training(titanic_split)
titanic_test <- testing(titanic_split)
titanic_split
```

```
## <Analysis/Assess/Total>
## <712/179/891>
```

```
dim(titanic_train) #number seems right to me
```

```
## [1] 712  12
```

```
dim(titanic_test) #number match the Assess number
```

```
## [1] 179  12
```

Question2: Fold the training data, k=10

```
titanic_rec <- recipe(survived ~ pclass + sex + age + sib_sp + parch + fare, data = titanic_train) %>%
  step_impute_linear(age) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_interact(terms =  ~ sex_male:fare) %>%
  step_interact(terms =  ~ age:fare)
```

```
set.seed(3534)
titanic_folds <- vfold_cv(titanic_train, v=10)
titanic_folds
```

```
## #  10-fold cross-validation
## # A tibble: 10 x 2
##    splits           id
##    <list>           <chr>
##  1 <split [640/72]> Fold01
##  2 <split [640/72]> Fold02
##  3 <split [641/71]> Fold03
##  4 <split [641/71]> Fold04
##  5 <split [641/71]> Fold05
##  6 <split [641/71]> Fold06
##  7 <split [641/71]> Fold07
##  8 <split [641/71]> Fold08
##  9 <split [641/71]> Fold09
## 10 <split [641/71]> Fold10
```

Question3:

1. We could have the chance to use all of our data by applying this method. Generally, compared to traditional fitting and testing model on the entire training set, we get to build K different models, so we are able to make predictions on all of our data.

   The second reason I could think of is that we can be more confident in our algorithm performance. When we do a single evaluation on our test set, we get only one result. This result may be because of chance or a biased test set for some reason. By training five (or ten) different models we can understand better what's going on.

2. If we did use the entire training set, the resampling method is validation set approach

Question4:

```r
#1
log_reg_titanic <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification")

log_wkf<-workflow() %>%
  add_model(log_reg_titanic) %>%
  add_recipe(titanic_rec)

#2
lda_mod_titanic <- discrim_linear() %>%
  set_mode("classification") %>%
  set_engine("MASS")

lda_wkf<-workflow() %>%
  add_model(lda_mod_titanic) %>%
  add_recipe(titanic_rec)

#3
qda_mod_titanic <- discrim_quad() %>%
  set_mode("classification") %>%
  set_engine("MASS")

qda_wkf<-workflow() %>%
  add_model(qda_mod_titanic) %>%
  add_recipe(titanic_rec)

#The total folds is going to be 30 (3x10)
```

Question5:

```r
log_fit_rs <- log_wkf %>%
  fit_resamples(titanic_folds)

lda_fit_rs <- lda_wkf %>%
  fit_resamples(titanic_folds)

qda_fit_rs <- qda_wkf %>%
  fit_resamples(titanic_folds)
```

Question7:

```r
log_metrics <- collect_metrics(log_fit_rs)
log_metrics
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.812    10 0.00917 Preprocessor1_Model1
## 2 roc_auc  binary     0.849    10 0.0108  Preprocessor1_Model1
```

```r
lda_metrics <- collect_metrics(lda_fit_rs)
lda_metrics
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.801    10 0.00933 Preprocessor1_Model1
## 2 roc_auc  binary     0.851    10 0.0102  Preprocessor1_Model1
```

```r
qda_metrics <- collect_metrics(qda_fit_rs)
qda_metrics
```

```
## # A tibble: 2 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.778    10  0.0191 Preprocessor1_Model1
## 2 roc_auc  binary     0.846    10  0.0124 Preprocessor1_Model1
```

The logistic regression analysis is the best model among these three because it has the highest accuracy mean.

Question7: fit the model, using training dataset

```r
final_fit <- fit(log_wkf, titanic_train)
final_fit
```

```
## == Workflow [trained] ==========================================================
## Preprocessor: Recipe
## Model: logistic_reg()
##
## -- Preprocessor ----------------------------------------------------------------
## 4 Recipe Steps
##
## * step_impute_linear()
## * step_dummy()
## * step_interact()
## * step_interact()
##
## -- Model -----------------------------------------------------------------------
##
## Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)
##
## Coefficients:
##     (Intercept)              age           sib_sp            parch
##      -4.3384398        0.0606187        0.4355448        0.2798873
##            fare        pclass_X2        pclass_X3         sex_male
##      -0.0063892        1.1642338        2.3270440        2.3718260
## sex_male_x_fare       age_x_fare
##       0.0136193       -0.0002814
##
## Degrees of Freedom: 711 Total (i.e. Null);  702 Residual
## Null Deviance:       948
## Residual Deviance: 618.4     AIC: 638.4
```

```r
predict(final_fit, new_data = titanic_train, type = "prob")
```

```
## # A tibble: 712 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.106     0.894
##  2    0.0786    0.921
##  3    0.290     0.710
##  4    0.0990    0.901
##  5    0.0116    0.988
##  6    0.776     0.224
##  7    0.0631    0.937
##  8    0.492     0.508
##  9    0.222     0.778
## 10    0.530     0.470
## # ... with 702 more rows
```

```r
train_acc <- augment(final_fit, new_data = titanic_train) %>%
  accuracy(truth = survived, estimate=.pred_class)
train_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.816
```

Question8:

```r
predict(final_fit, titanic_test, type = "prob")
```

```
## # A tibble: 179 x 2
##    .pred_Yes .pred_No
##        <dbl>    <dbl>
##  1    0.110    0.890
##  2    0.493    0.507
##  3    0.806    0.194
##  4    0.170    0.830
##  5    0.169    0.831
##  6    0.0440   0.956
##  7    0.162    0.838
##  8    0.563    0.437
##  9    0.909    0.0913
## 10    0.722    0.278
## # ... with 169 more rows
```

```r
test_acc <- augment(final_fit, new_data = titanic_test) %>%
  accuracy(truth = survived, estimate = .pred_class)
test_acc
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.782
```

```r
compare <- bind_cols(log_metrics[1,3], test_acc$.estimate)
```

```
## New names:
## * `` -> `...2`
```

```r
names(compare)[1] <- "average folds accuracy"
names(compare)[2] <- "testing accuracy"
compare
```

```
## # A tibble: 1 x 2
##   `average folds accuracy` `testing accuracy`
##                      <dbl>              <dbl>
## 1                    0.812              0.782
```