```
> library(titanic)    # loads titanic_train data frame
> library(caret)
> library(tidyverse)
> library(rpart)
>
> # 3 significant digits
> options(digits = 3)
>
> # clean the data - `titanic_train` is loaded with the titanic package
> titanic_clean <- titanic_train %>%
+     mutate(Survived = factor(Survived),
+            Embarked = factor(Embarked),
+            Age = ifelse(is.na(Age), median(Age, na.rm = TRUE), Age), # NA age to median age
+            FamilySize = SibSp + Parch + 1) %>%    # count family members
+     select(Survived,  Sex, Pclass, Age, Fare, SibSp, Parch, FamilySize, Embarked)
>
> dim(titanic_clean)
[1] 891   9
> head(titanic_clean)
  Survived    Sex Pclass Age  Fare SibSp Parch FamilySize Embarked
1        0   male      3  22  7.25     1     0          2        S
2        1 female      1  38 71.28     1     0          2        C
3        1 female      3  26  7.92     0     0          1        S
4        1 female      1  35 53.10     1     0          2        S
5        0   male      3  35  8.05     0     0          1        S
6        0   male      3  28  8.46     0     0          1        Q
>
>
>
> set.seed(42, sample.kind = "Rounding")
Warning message:
In set.seed(42, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> test_index <- createDataPartition(titanic_clean$Survived, times = 1, p = 0.2, list = FALSE)
> test_set <- titanic_clean[test_index,]
> train_set <- titanic_clean[-test_index,]
> dim(train_set)
[1] 712   9
> dim(test_set)
[1] 179   9
> mean(train_set$Survived == 1)
[1] 0.383
>
>
>
> set.seed(3, sample.kind = "Rounding")
Warning message:
In set.seed(3, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> guess <- sample(c(0,1), nrow(test_set), replace = TRUE)
> mean(guess == test_set$Survived)
[1] 0.475
>
>
>
> train_set %>% group_by(Sex) %>% summarize(survive = mean(Survived == 1), die = 1 - survive)
# A tibble: 2 x 3
  Sex    survive   die
  <chr>    <dbl> <dbl>
1 female   0.731 0.269
2 male     0.197 0.803
>
>
>
> sex_model <- ifelse(test_set$Sex == "female", 1, 0)
```

```
> mean(sex_model == test_set$Survived)
[1] 0.821
>
>
>
> train_set %>% group_by(Pclass) %>% summarize(survive = mean(Survived == 1), die = 1 - survive)
# A tibble: 3 x 3
  Pclass survive   die
   <int>   <dbl> <dbl>
1      1   0.619 0.381
2      2   0.5   0.5
3      3   0.242 0.758
>
>
>
> class_model <- ifelse(test_set$Pclass == 1, 1, 0)
> mean(class_model == test_set$Survived)
[1] 0.704
>
>
>
> train_set %>% group_by(Sex, Pclass) %>% summarize(survive = mean(Survived == 1), die = 1 - survive)
# A tibble: 6 x 4
# Groups:   Sex [2]
  Sex    Pclass survive    die
  <chr>   <int>   <dbl>  <dbl>
1 female      1   0.957 0.0435
2 female      2   0.919 0.0806
3 female      3   0.5   0.5
4 male        1   0.384 0.616
5 male        2   0.183 0.817
6 male        3   0.135 0.865
>
>
>
> sex_class_model <- ifelse(test_set$Sex == "female" & test_set$Pclass %in% 1:2, 1, 0)
> mean(sex_class_model == test_set$Survived)
[1] 0.821
>
>
>
> confusionMatrix(data = factor(sex_model), reference = test_set$Survived)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 96 18
         1 14 51

               Accuracy : 0.821
                 95% CI : (0.757, 0.874)
    No Information Rate : 0.615
    P-Value [Acc > NIR] : 1.72e-09

                  Kappa : 0.619

 Mcnemar's Test P-Value : 0.596

            Sensitivity : 0.873
            Specificity : 0.739
         Pos Pred Value : 0.842
         Neg Pred Value : 0.785
             Prevalence : 0.615
         Detection Rate : 0.536
   Detection Prevalence : 0.637
```

```
      Balanced Accuracy : 0.806

       'Positive' Class : 0

> confusionMatrix(data = factor(class_model), reference = test_set$Survived)
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 94 37
         1 16 32

               Accuracy : 0.704
                 95% CI : (0.631, 0.77)
    No Information Rate : 0.615
    P-Value [Acc > NIR] : 0.00788

                  Kappa : 0.337

 Mcnemar's Test P-Value : 0.00601

            Sensitivity : 0.855
            Specificity : 0.464
         Pos Pred Value : 0.718
         Neg Pred Value : 0.667
             Prevalence : 0.615
         Detection Rate : 0.525
   Detection Prevalence : 0.732
      Balanced Accuracy : 0.659

       'Positive' Class : 0

> confusionMatrix(data = factor(sex_class_model), reference = test_set$Survived)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 109  31
         1   1  38

               Accuracy : 0.821
                 95% CI : (0.757, 0.874)
    No Information Rate : 0.615
    P-Value [Acc > NIR] : 1.72e-09

                  Kappa : 0.589

 Mcnemar's Test P-Value : 2.95e-07

            Sensitivity : 0.991
            Specificity : 0.551
         Pos Pred Value : 0.779
         Neg Pred Value : 0.974
             Prevalence : 0.615
         Detection Rate : 0.609
   Detection Prevalence : 0.782
      Balanced Accuracy : 0.771

       'Positive' Class : 0

>
>
>
> F_meas(data = factor(sex_model), reference = test_set$Survived)
[1] 0.857
```

```
> F_meas(data = factor(class_model), reference = test_set$Survived)
[1] 0.78
> F_meas(data = factor(sex_class_model), reference = test_set$Survived)
[1] 0.872
>
```