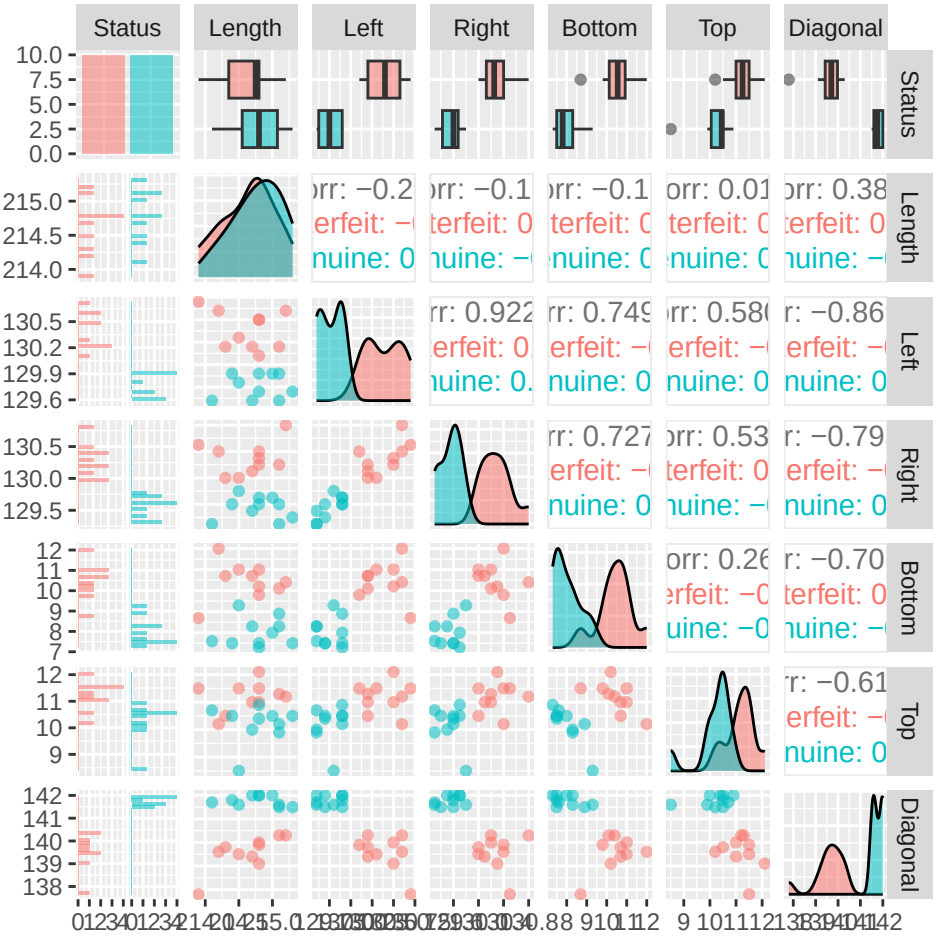
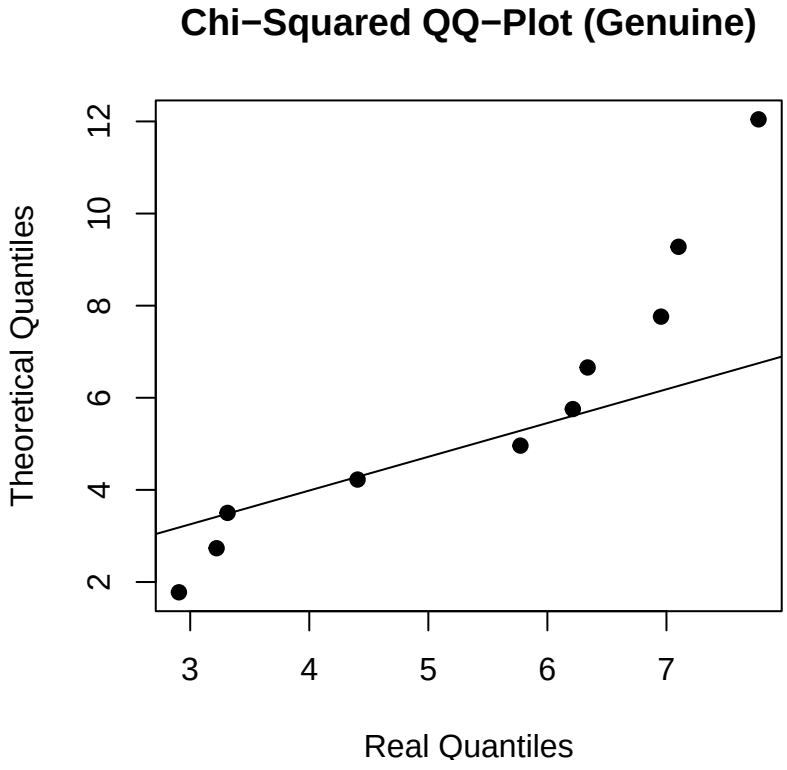
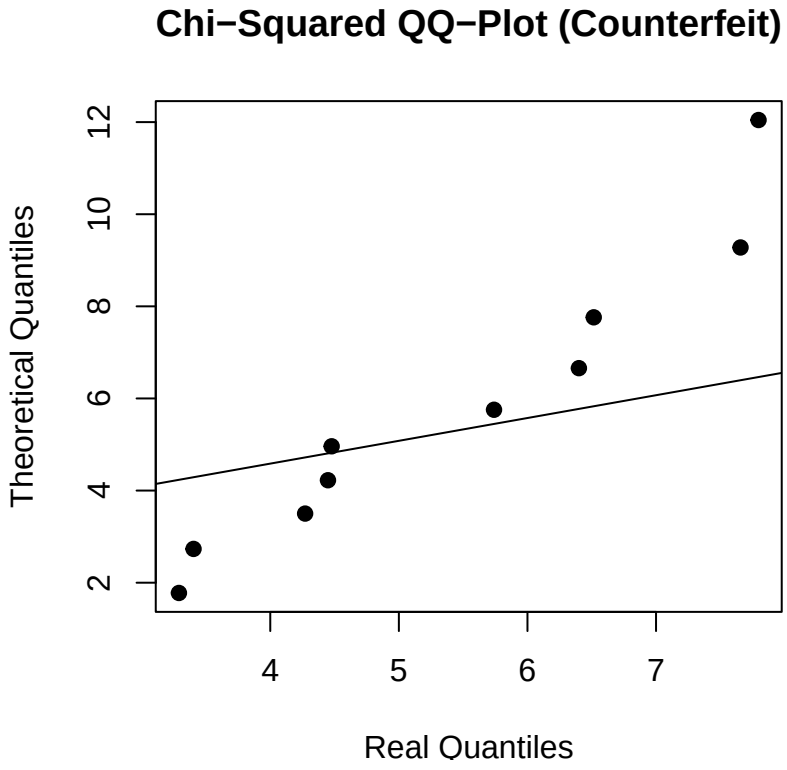
Swiss Banknote Data PCA and LDA

Carson Slater

Checking Multivariate Normality




Chi−Squared QQ−Plot (Genuine)
Theoretical Quantiles
Real Quantiles
2
4
6
8
10
12
3
4
5
6
7


Chi−Squared QQ−Plot (Counterfeit)
Theoretical Quantiles
Real Quantiles
2
4
6
8
10
12
4
5
6
7

For the nost part, the mahalanobis distance indicates that it strongly resembles a χ^2_6 distribution for both the genuine and counterfeit samples. We elect to proceed normally under the assumption of multivariate normality, which is useful for principle component analysis.

Initial PCA on `banknote` Data To Find PC with Maximal Separation of Means

```
banknote_standard <- scale(sample[,-1])

pca_bank <- princomp(banknote_standard, cor = TRUE)

scores <- summary(pca_bank, loadings = FALSE)$scores
scores <- cbind(sample[,1], scores)

c_scores <- scores |> dplyr::filter(Status == "counterfeit")
g_scores <- scores |> dplyr::filter(Status == "genuine")

(mahalanobis_dist <- (colMeans(c_scores[,-1]) - colMeans(g_scores[,-1]))^2/eigen(cov(banknote
```

```
     Comp.1      Comp.2      Comp.3      Comp.4      Comp.5      Comp.6
3.703684477 0.020775551 0.016301385 0.100670985 0.001105549 0.065273613
```

The mahalanobis distance is for the first principle component is the largest by far, so we elect to only use the first principle component in our LDA.

PCA + LDA on `banknote` Data

```
pca_rec <- recipe(Status ~ ., data = sample) |>
  step_normalize(all_numeric_predictors()) |>
  step_pca(all_numeric_predictors())

prep_pca <- prep(pca_rec, training = sample)

baked_pca <- bake(prep_pca, sample)

model_lda <- train(Status ~ .,
```

```
                    data = baked_pca[,1:2],
                    method = "lda",
                    trControl = trainControl("cv",
                                             number = 10)
                    )

model_lda
```

```
Linear Discriminant Analysis

20 samples
 1 predictor
 2 classes: 'counterfeit', 'genuine'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18, 18, 18, 18, 18, 18, ...
Resampling results:

  Accuracy  Kappa
  1         1
```

PCA + LDA on `iris` Data

```
data("iris")

iris_standard <- scale(iris[,-5])

pca_iris <- princomp(iris_standard, cor = TRUE)

scores <- summary(pca_iris, loadings = FALSE)$scores
scores <- cbind(iris[,5], scores) |> data.frame()

vir_scores <- scores |> dplyr::filter(V1 == 1)
ver_scores <- scores |> dplyr::filter(V1 == 2)

(mahalanobis_dist <- (colMeans(vir_scores[,-1]) - colMeans(ver_scores[,-1]))^2/eigen(cov(iris
```

```
    Comp.1      Comp.2      Comp.3      Comp.4
```

```
2.53724224 0.77030841 0.01935083 0.11440229
```

The first two components seems to maximally separate the means for the discriminant analysis, so we elect to only use those two.

```
iris_pca_rec <- recipe(Species ~ ., data = iris) |>
  step_normalize(all_numeric_predictors()) |>
  step_pca(all_numeric_predictors())

iris_prep_pca <- prep(iris_pca_rec, training = iris)

iris_baked_pca <- bake(iris_prep_pca, iris)

iris_model_lda <- train(Species ~ .,
                    data = iris_baked_pca[,1:3],
                    method = "lda",
                    trControl = trainControl("cv",
                                             number = 10)
                    )

iris_model_lda
```

```
Linear Discriminant Analysis

150 samples
  2 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results:

  Accuracy   Kappa
  0.9133333  0.87
```

banknote Linear Discriminant Analysis Without PCA

```
model_lda <- train(Status ~ .,
                   data = sample,
                   method = "lda",
                   trControl = trainControl("cv",
                                            number = 10)
                   )

model_lda
```

```
Linear Discriminant Analysis

20 samples
 6 predictor
 2 classes: 'counterfeit', 'genuine'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18, 18, 18, 18, 18, 18, ...
Resampling results:

  Accuracy  Kappa
  1         1
```

banknote Quadratic Discriminant Analysis

```
model_qda <- train(Status ~ .,
                   data = sample,
                   method = "qda",
                   trControl = trainControl("cv",
                                            number = 10)
                   )

model_qda
```

```
Quadratic Discriminant Analysis

20 samples
 6 predictor
```

```
 2 classes: 'counterfeit', 'genuine'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 18, 18, 18, 18, 18, 18, ...
Resampling results:

  Accuracy  Kappa
  0.95      0.9
```