Ian Mulchrone

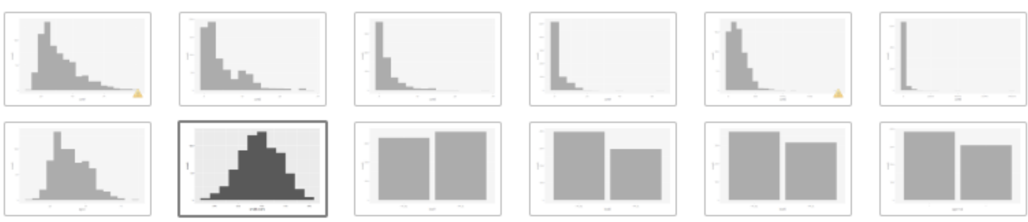
DSC 441

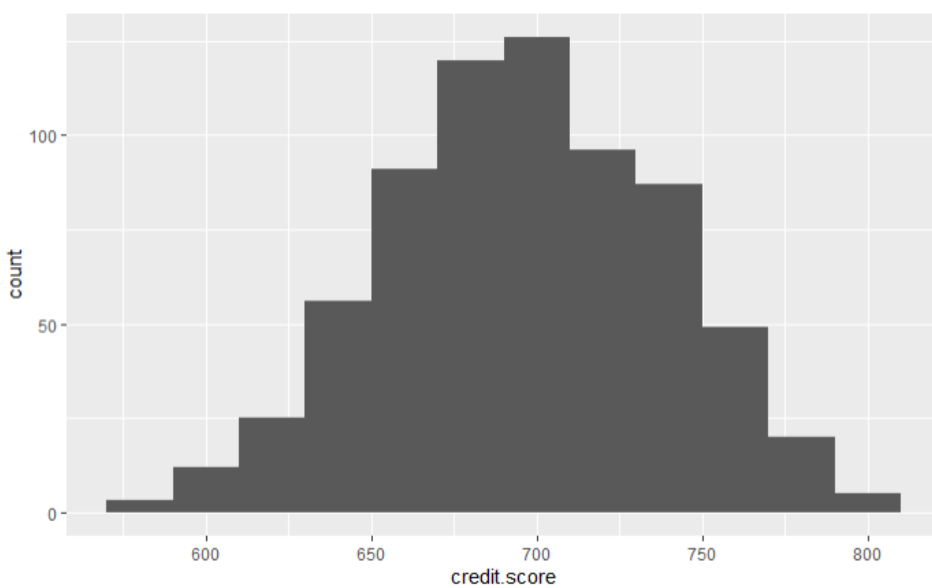
Homework 2

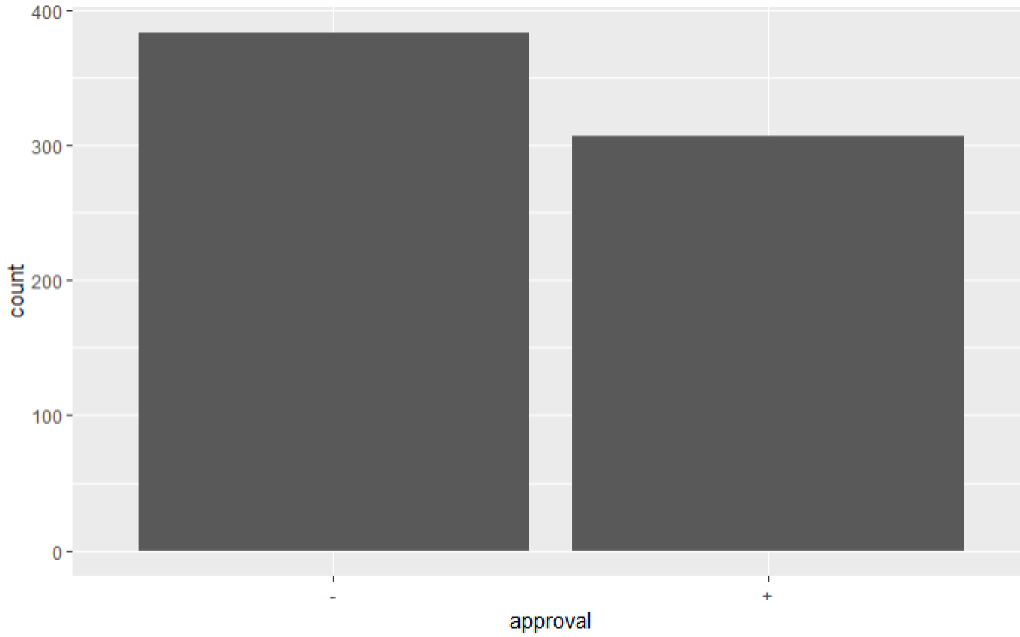
Problem 1

a.

```
      ...1            cont1            cont2            cont3           bool1            bool2        
 Min.   :  1.0   Min.   :13.75   Min.   : 0.000   Min.   : 0.000   Mode :logical   Mode :logical  
 1st Qu.:173.2   1st Qu.:22.60   1st Qu.: 1.000   1st Qu.: 0.165   FALSE:329       FALSE:395      
 Median :345.5   Median :28.46   Median : 2.750   Median : 1.000   TRUE :361       TRUE :295      
 Mean   :345.5   Mean   :31.57   Mean   : 4.759   Mean   : 2.223                                  
 3rd Qu.:517.8   3rd Qu.:38.23   3rd Qu.: 7.207   3rd Qu.: 2.625                                  
 Max.   :690.0   Max.   :80.25   Max.   :28.000   Max.   :28.500                                  
                 NA's   :12                                                                       
     cont4          bool3             cont5            cont6             approval          credit.score  
 Min.   : 0.0   Mode :logical   Min.   :   0   Min.   :     0.0   Length:690         Min.   :583.7  
 1st Qu.: 0.0   FALSE:374       1st Qu.:  75   1st Qu.:     0.0   Class :character   1st Qu.:666.7  
 Median : 0.0   TRUE :316       Median : 160   Median :     5.0   Mode  :character   Median :697.3  
 Mean   : 2.4                   Mean   : 184   Mean   :  1017.4                      Mean   :696.4  
 3rd Qu.: 3.0                   3rd Qu.: 276   3rd Qu.:   395.5                      3rd Qu.:726.4  
 Max.   :67.0                   Max.   :2000   Max.   :100000.0                      Max.   :806.0  
                                NA's   :13                                                        
      ages      
 Min.   :11.00  
 1st Qu.:31.00  
 Median :38.00  
 Mean   :39.67  
 3rd Qu.:48.00  
 Max.   :84.00  
```



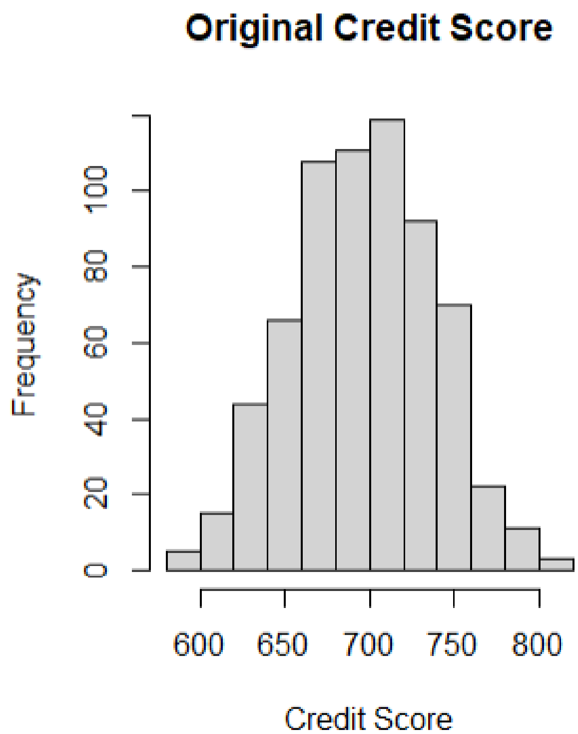


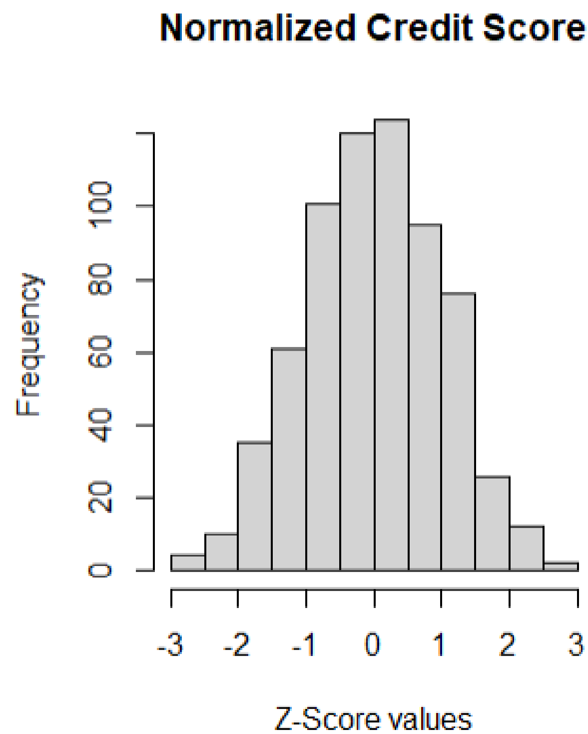


Looking at the distributions of the numerical variables, credit score is the only one that displays a normal distribution, while the rest are all right skewed, with some having extreme outliers on the right end. For the Boolean variables, the bar graphs don't show any extreme disparity between categories, so we don't need to worry about over or under representation in our data.

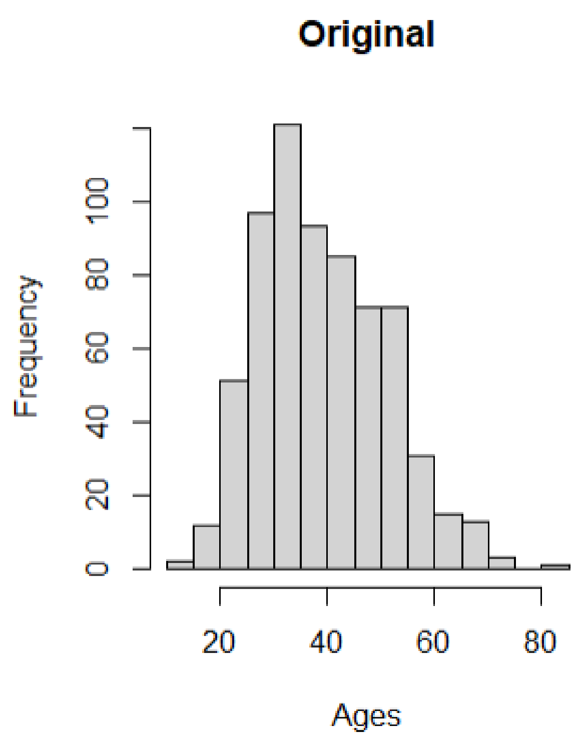
b. We will apply z-score normalization to credit score. This will change the values of each credit score to its corresponding standard deviation value from the mean. Since the distribution is already normal, this transformation will not change its distribution at all.
   Min-max normalization will be applied to ages. This will change the scale for the distributions so that all values will fall between 0 and 1. It should not meaningfully affect the distribution and will remain slightly right skewed.
   Decimal scaling normalization will be applied to cont2. Since we don't have any negative values in the dataset, this type of normalization doesn't make much sense and so won't have any effect on the very right skewed distribution.

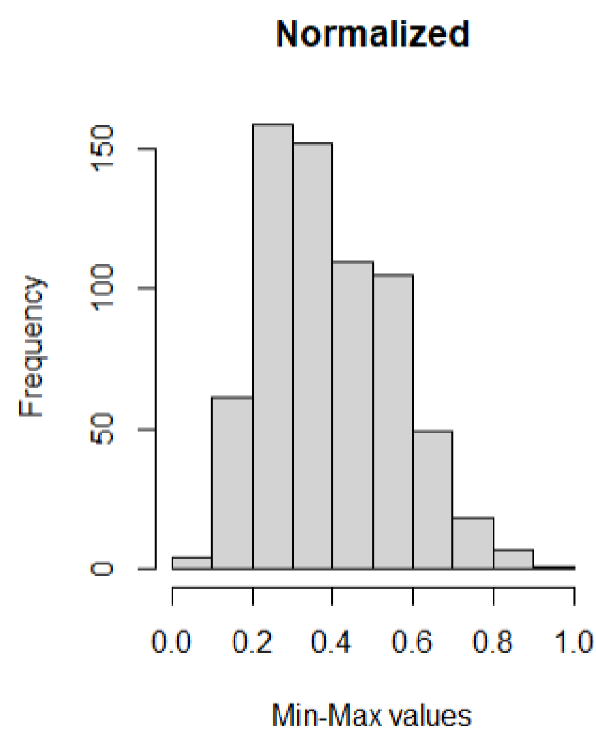
c.



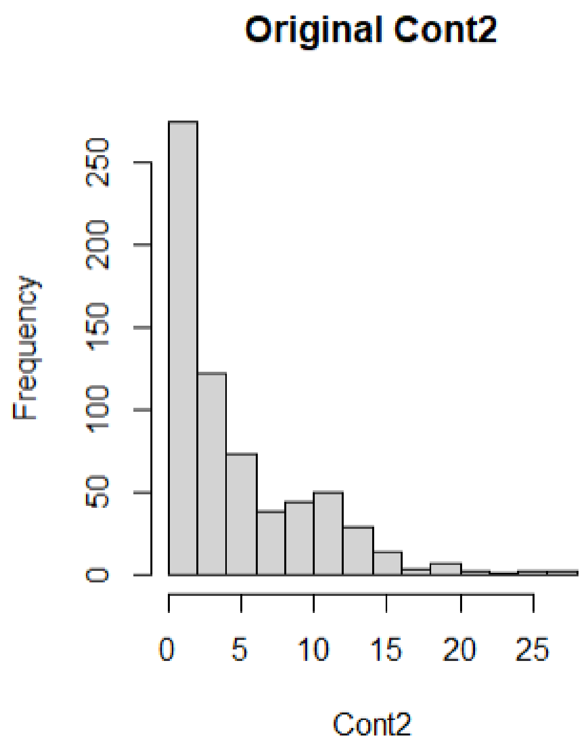


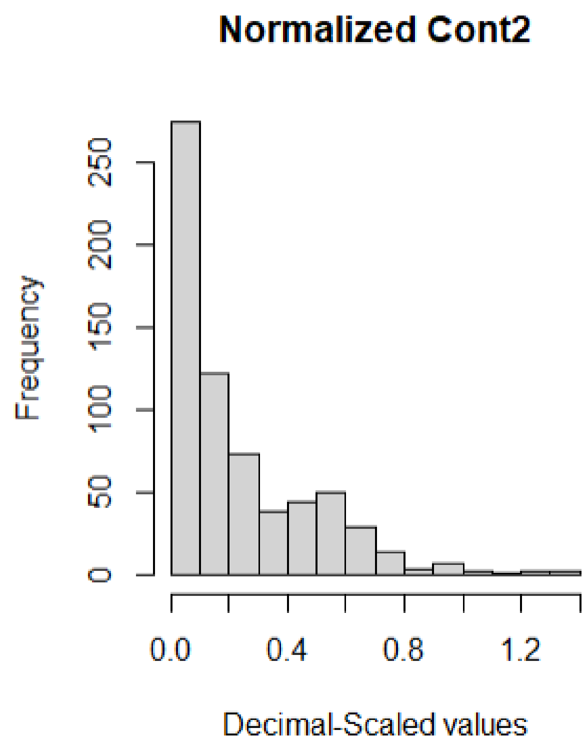
The distribution for credit score remains normal.





Distribution has not meaningfully changed.





Distribution is the exact same.

d. I've chosen to bin using the credit score variable using the low, medium, and high values. Since it is on a normal scale, it makes sense to split it up into three categories, with the medium values being within one standard deviation to the mean, and the low and high values being on the tails. Looking at the distribution, I've estimated that the medium values with one standard deviation are between 660 and 740. So, anything lower than 660 will be low and credit scores above 740 will be high.

bool3 <lgl>	cont5 <dbl>	cont6 <dbl>	approval <chr>	credit.score <dbl>	ages <dbl>	cont2_ds <dbl>	credit.score_ds <dbl>	ages_ds <dbl>	cs_bins <fctr>
FALSE	202	0	+	664.60	42	0.00000	22.15333	2.10	medium
FALSE	43	560	+	693.88	54	0.22300	23.12933	2.70	medium
FALSE	280	824	+	621.82	29	0.02500	20.72733	1.45	low
TRUE	100	3	+	653.97	58	0.07700	21.79900	2.90	low
FALSE	120	0	+	670.26	65	0.28125	22.34200	3.25	medium
TRUE	360	0	+	672.16	61	0.20000	22.40533	3.05	medium

6 rows | 8-17 of 17 columns

e.

cont4 <dbl>	bool3 <lgl>	cont5 <dbl>	cont6 <dbl>	approval <chr>	credit.score <dbl>	ages <dbl>	cont2_ds <dbl>	cs_bins <fctr>	cs_cat <dbl>
1	FALSE	202	0	+	664.60	42	0.00000	medium	1
6	FALSE	43	560	+	693.88	54	0.22300	medium	1
0	FALSE	280	824	+	621.82	29	0.02500	low	0
5	TRUE	100	3	+	653.97	58	0.07700	low	0
0	FALSE	120	0	+	670.26	65	0.28125	medium	1
0	TRUE	360	0	+	672.16	61	0.20000	medium	1

I added a new variable cs_cat that uses numerical categories for low, medium, and high with 0, 1, and 2 repecetively. I chose to do it this way instead of dummy variables because there is an inherent ranking with credit scores, with lower values being worse and higher values being better. So, ranking the categories as 0, 1, and 2 is representative of what we are trying to determine with low, medium, and high credit score categories.

Problem 2

a.

```
Support Vector Machines with Linear Kernel

666 samples
 16 predictor
  2 classes: '-', '+'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 600, 600, 600, 599, 599, 600, ...
Resampling results:

  Accuracy  Kappa
  1         1

Tuning parameter 'C' was held constant at a value of 1
```

Accuracy is 100%.

b.

```
Support Vector Machines with Linear Kernel

666 samples
 16 predictor
  2 classes: '-', '+'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 533, 533, 532, 533, 533
Resampling results across tuning parameters:

  C             Accuracy   Kappa
  1.000000e-05  0.5510493  0.0000000
  3.162278e-05  0.5510493  0.0000000
  1.000000e-04  0.5510493  0.0000000
  3.162278e-04  0.6532039  0.2439333
  1.000000e-03  0.9639434  0.9271553
  3.162278e-03  1.0000000  1.0000000
  1.000000e-02  1.0000000  1.0000000
  3.162278e-02  1.0000000  1.0000000
  1.000000e-01  1.0000000  1.0000000
  3.162278e-01  1.0000000  1.0000000
  1.000000e+00  1.0000000  1.0000000
  3.162278e+00  1.0000000  1.0000000
  1.000000e+01  1.0000000  1.0000000
  3.162278e+01  1.0000000  1.0000000
  1.000000e+02  1.0000000  1.0000000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was C = 0.003162278.
```

Parameter chosen was C = 0.003162278 and the accuracy is 100%.

c. When using cross-validation, the folds could be different which could result in a slightly different model. So, even if you kept the default value C = 1, the accuracy could be different based on the randomness of the train-test split used when building the model.

Problem 3

a.

	height <dbl>	mass <dbl>	hair_colorauburn, white <dbl>	hair_colorblack <dbl>	hair_colorblond <dbl>	hair_colorbrown <dbl>
1	172	77	0	0	1	0
2	202	136	0	0	0	0
3	150	49	0	0	0	1
4	178	120	0	0	0	0
5	165	75	0	0	0	1
6	183	84	0	1	0	0

6 rows | 1-7 of 66 columns

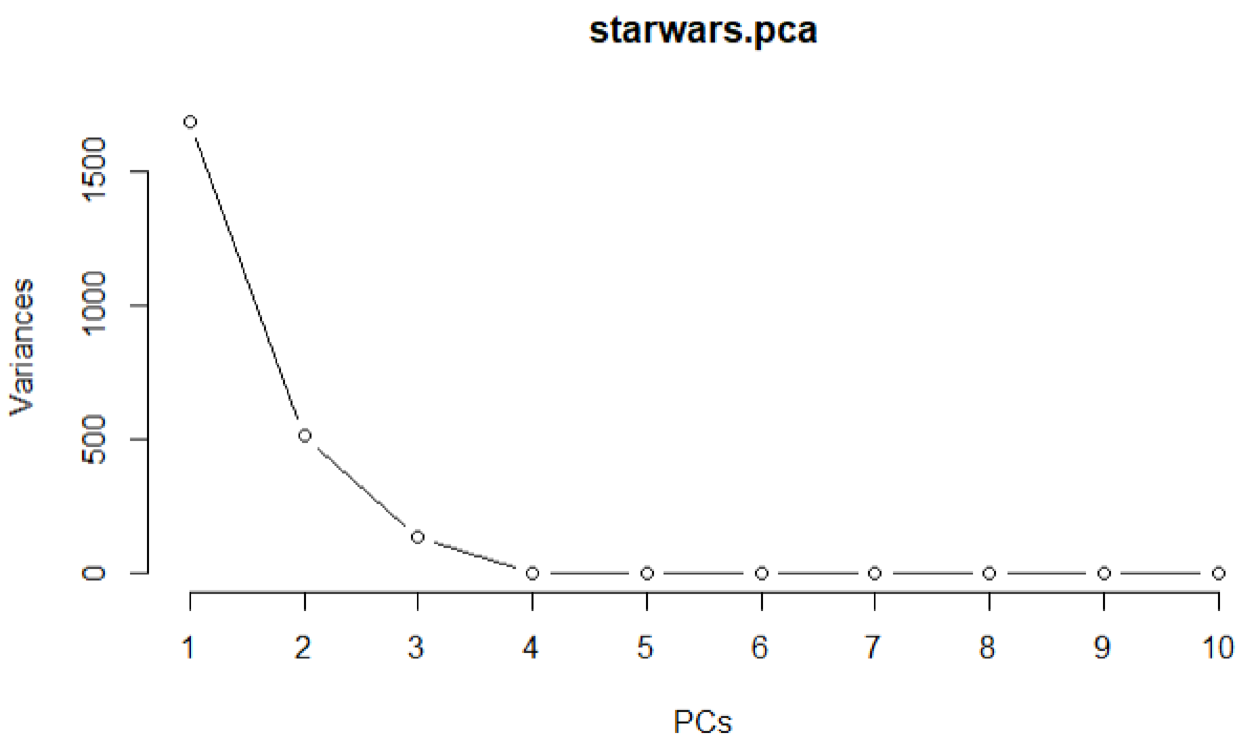
b.

```
29 samples
 9 predictor
 2 classes: 'feminine', 'masculine'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 29, 29, 29, 29, 29, 29, ...
Resampling results:

  Accuracy   Kappa
  0.8562954  0.5869239

Tuning parameter 'C' was held constant at a value of 1
```

Accuracy is 85.63%.

c.

```
Importance of components:
                           PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8    PC9    PC10
Standard deviation     41.1152 22.6543 11.5834 0.78553 0.73967 0.59403 0.57852 0.51620 0.4766 0.42229
Proportion of Variance  0.7219  0.2192  0.0573 0.00026 0.00023 0.00015 0.00014 0.00011 0.0001 0.00008
Cumulative Proportion   0.7219  0.9411  0.9984 0.99863 0.99886 0.99901 0.99916 0.99927 0.9994 0.99944
                          PC11    PC12    PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20
Standard deviation     0.38431 0.37487 0.36960 0.35334 0.34330 0.32559 0.31189 0.29516 0.27246 0.25433
Proportion of Variance 0.00006 0.00006 0.00006 0.00005 0.00005 0.00005 0.00004 0.00004 0.00003 0.00003
Cumulative Proportion  0.99951 0.99957 0.99963 0.99968 0.99973 0.99977 0.99982 0.99985 0.99988 0.99991
                          PC21    PC22    PC23    PC24    PC25    PC26    PC27    PC28      PC29
Standard deviation     0.24201 0.20171 0.19847 0.16261 0.15702 0.11978 0.03711 0.01792 3.836e-15
Proportion of Variance 0.00003 0.00002 0.00002 0.00001 0.00001 0.00001 0.00000 0.00000 0.000e+00
Cumulative Proportion  0.99994 0.99995 0.99997 0.99998 0.99999 1.00000 1.00000 1.00000 1.000e+00
```



	PC1 <dbl>	PC2 <dbl>	PC3 <dbl>	gender <chr>
1	-0.2119596	1.7798166	-0.1805155	masculine
2	2.6062413	0.7266943	0.0536266	masculine
3	-3.5052503	0.3029799	-0.5375610	feminine
4	0.3975989	1.7973980	0.7166245	masculine
5	-2.1157926	0.6934936	1.1160075	feminine
6	-0.7175118	1.4161888	-0.5011268	masculine

6 rows

After the second PC we have over 94% of the cumulative proportion of variance explained which would be good enough. However, there is a significant increase with PC3 before the variance drops off. So, I have decided to include 3 PCs in the model.

d. I first partitioned the data into a 70-30 train-test split. With a small dataset like this, I wanted to make sure I had an adequate number of test observations and I felt that 80-20 would be too few. This way, we have 22 in the training set and 9 in the test set. Using the training set, I created a model using a 5-fold cross validation partition. The model has a 92% accuracy.

```
Support Vector Machines with Linear Kernel

22 samples
 3 predictor
 2 classes: 'feminine', 'masculine'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 18, 18, 17, 17, 18
Resampling results:

  Accuracy  Kappa
  0.92      0.6

Tuning parameter 'C' was held constant at a value of 1
```

Now, we use our model to check it accuracy on the test set. We see an 85% accuracy on the test set, with one false negative in the feminine category. Because the data had a higher percentage of masculine genders, it makes sense that the test shows some bias towards masculine. We will need more feminine data to remedy this problem.

```
Confusion Matrix and Statistics

           Reference
Prediction  feminine masculine
  feminine         1         0
  masculine        1         5

               Accuracy : 0.8571
                 95% CI : (0.4213, 0.9964)
    No Information Rate : 0.7143
    P-Value [Acc > NIR] : 0.3605

                  Kappa : 0.5882

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.5000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.8333
             Prevalence : 0.2857
         Detection Rate : 0.1429
   Detection Prevalence : 0.1429
      Balanced Accuracy : 0.7500

       'Positive' Class : feminine
```

e. PCA reduces complexity by immediately identifying the principal components that capture the most variance in the model, reducing both the number of variables and the time it takes to identify the variables with most significance. Especially in this scenario, where we had so many dummy variables from the categorical columns, we went from over 60 variables down to 3 with just one step. We could have reduced it to 2 variables to minimize computing power, but by using PCA we can make informed decisions about our model and identify the positives and negatives of adding more variables to the model.