Project2

Rose Hedderman EID: rrh2298

4/17/2021

```
knitr::opts_chunk$set(warning = FALSE)
options(warn=-1)
```

Project 2

Introduction.

Introduce your dataset and each of your variables (or just the main variables if you have many). How was the data collected? How many observations? Did you have to tidy the data? Why are you interested in exploring this dataset? What do you expect to find?

```
# import needed libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
--- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.3
## v tibble  3.0.5     v stringr 1.4.0
## v tidyr   1.1.2     v forcats 0.5.0
## v readr   1.3.1
```

```
## -- Conflicts ------------------------------------------------------------------------------------ ti
dyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(cluster)
options(warn=-1)
# import dataset
models <- read.csv("C:/Users/roseh/OneDrive/Desktop/Spring 2021/SDS 348/best_results_test_105_models (1).cs
v", stringsAsFactors=FALSE)
# view the first 6 rows of the dataset 'models'
head(`models`)
```

```
##   biophysical_model_type NN_depth NN_width NN_input_time_window NN_model_type
## 1                   AMPA        1       64                   35           FCN
## 2                   AMPA        1       32                   39           FCN
## 3                   AMPA        1      128                   43           FCN
## 4                   AMPA        1      256                   45           FCN
## 5                   AMPA        1      128                   54           FCN
## 6                   AMPA        1      256                   55           FCN
##   spikes.D.prime spikes.AUC spikes.AUC...1..FP soma.explained.variance..
## 1       3.340230  0.9909092          0.4345988                  95.02454
## 2       3.206526  0.9883158          0.3806042                  94.16525
## 3       3.365291  0.9913348          0.4361554                  95.06362
## 4       3.354546  0.9911545          0.4289646                  95.08548
## 5       3.337530  0.9908623          0.4333841                  95.02461
## 6       3.350060  0.9910783          0.4292489                  95.23931
##   soma.RMSE  soma.MAE spikes.TP...0.1..FP spikes.TP...0.25..FP
## 1 0.5910565 0.3309638           0.1209919            0.2718370
## 2 0.6400739 0.3746568           0.1031433            0.2367714
## 3 0.5887358 0.3312969           0.1124625            0.2753120
## 4 0.5874254 0.3253216           0.1021955            0.2674143
## 5 0.5910526 0.3300470           0.1132523            0.2705734
## 6 0.5781574 0.3201280           0.1108830            0.2753120
##   spikes.AUC.std.of.subsets soma.explained.variance...std.of.subsets
## 1              0.0003991853                               0.08707119
## 2              0.0005286728                               0.10242624
## 3              0.0003469035                               0.09148894
## 4              0.0003150930                               0.10221177
## 5              0.0003402101                               0.06044799
## 6              0.0003360585                               0.09098617
##   NN_num_train_samples NN_unique_train_files
## 1              1873920                   432
## 2               855360                   426
## 3              2304000                   432
## 4              1088640                   430
## 5              1536000                   432
## 6              1347840                   432
##                                                                                                      full.mod
el.filename
## 1  AMPA_FCN__DxWxT_1x64x35__2019-10-20__11_55__samples_1873920__LogLoss_train_43_valid_48__ID_60131_eval
uation_test
## 2   AMPA_FCN__DxWxT_1x32x39__2019-10-27__12_58__samples_855360__LogLoss_train_43_valid_50__ID_85899_eval
uation_test
## 3 AMPA_FCN__DxWxT_1x128x43__2019-09-09__17_50__samples_2304000__LogLoss_train_39_valid_46__ID_59608_eval
uation_test
## 4 AMPA_FCN__DxWxT_1x256x45__2019-11-11__18_17__samples_1088640__LogLoss_train_42_valid_47__ID_82086_eval
uation_test
## 5 AMPA_FCN__DxWxT_1x128x54__2019-10-27__08_30__samples_1536000__LogLoss_train_39_valid_48__ID_66716_eval
uation_test
## 6  AMPA_FCN__DxWxT_1x256x55__2019-11-03__23_42__samples_1347840__LogLoss_train_37_valid_49__ID_1816_eval
uation_test
```

```
# view column names to rename columns
```

EDA.

Explore your main variables by producing univariate/bivariate statistics and graphs. In particular, investigate relationships that you are going to test about with MANOVA and a randomization test, and variables included in your regression models. For example, you could create a correlation matrix with univariate/bivariate graphs and correlation coefficients

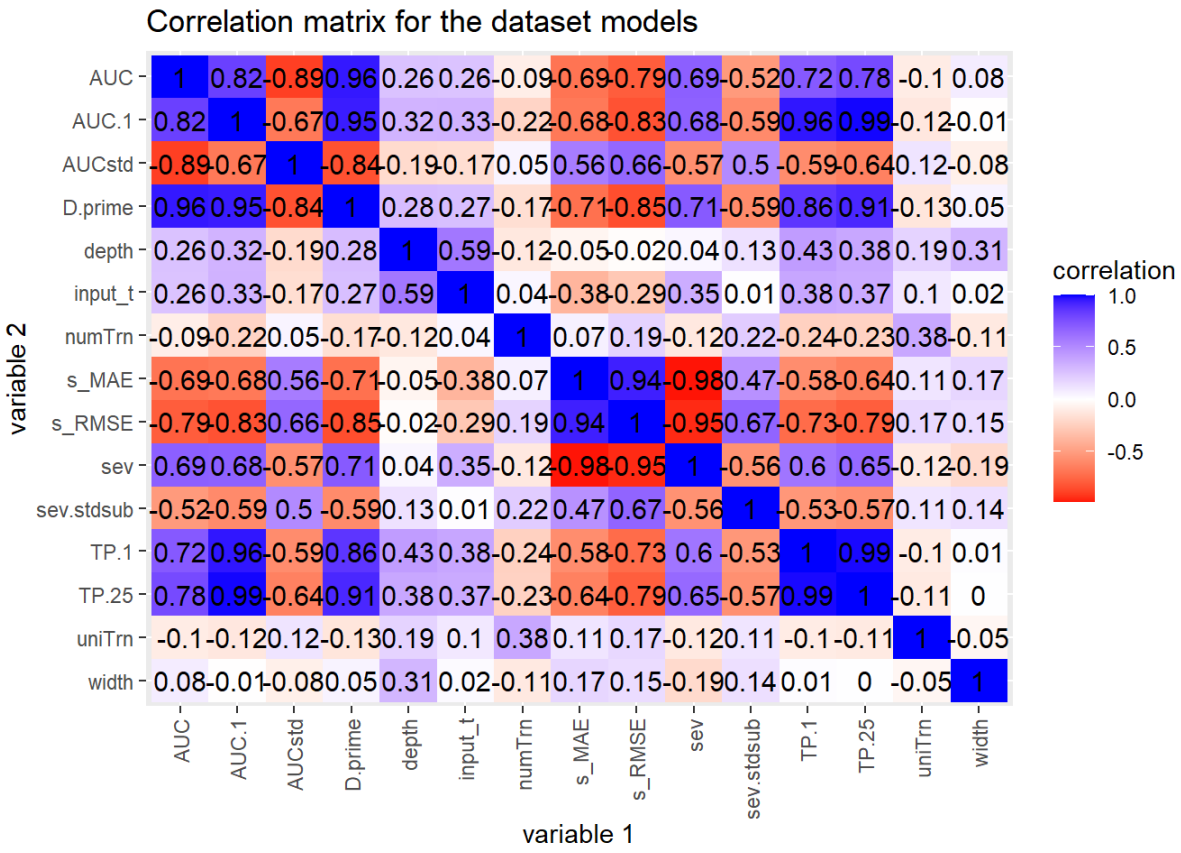
```
# graphs - correlation matrix
# view column names to rename columns so correlation matrix dimensions fit
colnames(models)
```

```
##  [1] "biophysical_model_type"
##  [2] "NN_depth"
##  [3] "NN_width"
##  [4] "NN_input_time_window"
##  [5] "NN_model_type"
##  [6] "spikes.D.prime"
##  [7] "spikes.AUC"
##  [8] "spikes.AUC...1..FP"
##  [9] "soma.explained.variance.."
## [10] "soma.RMSE"
## [11] "soma.MAE"
## [12] "spikes.TP...0.1..FP"
## [13] "spikes.TP...0.25..FP"
## [14] "spikes.AUC.std.of.subsets"
## [15] "soma.explained.variance...std.of.subsets"
## [16] "NN_num_train_samples"
## [17] "NN_unique_train_files"
## [18] "full.model.filename"
```

```
models <- models %>%
    # rename variables to fix dimensions of correlation matrix
  rename("depth" = NN_depth, "width" = NN_width, "input_t" = NN_input_time_window,
         "D.prime" = spikes.D.prime, "AUC" = spikes.AUC, "AUC.1" = spikes.AUC...1..FP,
         "sev" = soma.explained.variance.., "s_RMSE" = soma.RMSE, "s_MAE" = soma.MAE,
         "TP.1" = spikes.TP...0.1..FP, "TP.25" = spikes.TP...0.25..FP,
         "AUCstd" = spikes.AUC.std.of.subsets, "sev.stdsub" = soma.explained.variance...std.of.subsets,
         "numTrn" = NN_num_train_samples, "uniTrn" = NN_unique_train_files)
# create a correlation matrix with univariate/bivariate graphs and correlation coefficients
nummodels <- models %>%
  select(-biophysical_model_type, -full.model.filename, -NN_model_type)



nummodels <- nummodels %>%
  scale %>%
  as.data.frame
head(nummodels)
```

```
##       depth      width    input_t     D.prime         AUC      AUC.1       sev
## 1 -1.245682 -0.4664540 -0.8459959  0.06779699  0.28070963 -0.1851356 0.3185441
## 2 -1.245682 -0.8828422 -0.7652968 -0.31789684 -0.01779367 -0.6200840 0.1117246
## 3 -1.245682  0.3663225 -0.6845976  0.14008987  0.32968913 -0.1725963 0.3279498
## 4 -1.245682  2.0318755 -0.6442481  0.10909250  0.30893973 -0.2305212 0.3332110
## 5 -1.245682  0.3663225 -0.4626750  0.06000700  0.27530812 -0.1949202 0.3185611
## 6 -1.245682  2.0318755 -0.4425002  0.09615195  0.30016609 -0.2282309 0.3702380
##       s_RMSE      s_MAE       TP.1      TP.25     AUCstd sev.stdsub     numTrn
## 1 -0.4220773 -0.3952568 -0.4619371 -0.3919605 -0.1887656 -0.3696419  1.3480573
## 2 -0.2122866 -0.1180036 -0.7025039 -0.6941841  0.1551264 -0.2332559 -0.2116784
## 3 -0.4320099 -0.3931435 -0.5768982 -0.3620104 -0.3276153 -0.3304027  2.0066451
## 4 -0.4376183 -0.4310592 -0.7152774 -0.4300788 -0.4120974 -0.2351609  0.1455466
## 5 -0.4220941 -0.4010746 -0.5662537 -0.4028514 -0.3453915 -0.6061135  0.8305955
## 6 -0.4772847 -0.4640152 -0.5981873 -0.3620104 -0.3564175 -0.3348685  0.5424634
##      uniTrn
## 1 0.6885882
## 2 0.5847312
## 3 0.6885882
## 4 0.6539692
## 5 0.6885882
## 6 0.6885882
```

```
# Find the correlations among the 10 disciplines
cor(nummodels, use = "pairwise.complete.obs") %>%
  # Save as a data frame
  as.data.frame %>%
  # Convert row names to an explicit variable
  rownames_to_column %>%
  # Pivot so that all correlations appear in the same column
  pivot_longer(-1, names_to = "other_var", values_to = "correlation") %>%
  ggplot(aes(rowname, ordered(other_var, levels = rev(sort(unique(other_var)))), fill=correlation)) +
  # Heatmap with geom_tile
  geom_tile() +
  # Change the scale to make the middle appear neutral
  scale_fill_gradient2(low="red",mid="white",high="blue") +
  # Overlay values
  geom_text(aes(label = round(correlation,2)), color = "black", size = 4) +
  # Give title and labels
  labs(title = "Correlation matrix for the dataset models", x = "variable 1", y = "variable 2") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



MANOVA

Perform a MANOVA to test whether any of your numeric variables (or a subset of them, if including them all doesn't make sense) show a mean difference across levels of one of your categorical variables. If significant, perform univariate ANOVAs to find response(s) showing a mean difference across groups, and perform post-hoc t tests to find which groups differ. Discuss the number of tests you have performed, calculate the probability of at least one type I error, and adjust the significance level accordingly (Bonferroni correction) before discussing significant differences. Briefly discuss assumptions and whether or not they are likely to have been met.

```
##                          Df Pillai approx F num Df den Df    Pr(>F)
## biophysical_model_type    2 1.1522   12.775     20    188 < 2.2e-16 ***
## Residuals               102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##  Response AUC :
##                          Df    Sum Sq    Mean Sq F value    Pr(>F)
## biophysical_model_type   2 0.0038303 0.00191514  48.592 1.502e-15 ***
## Residuals              102 0.0040201 0.00003941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response AUC.1 :
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## biophysical_model_type   2 0.96990 0.48495  78.164 < 2.2e-16 ***
## Residuals              102 0.63283 0.00620
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response AUCstd :
##                          Df     Sum Sq    Mean Sq F value    Pr(>F)
## biophysical_model_type   2 5.3647e-06 2.6823e-06  29.167 9.599e-11 ***
## Residuals              102 9.3803e-06 9.1960e-08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response D.prime :
##                          Df Sum Sq Mean Sq F value    Pr(>F)
## biophysical_model_type   2 7.7727  3.8864  83.894 < 2.2e-16 ***
## Residuals              102 4.7251  0.0463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response s_MAE :
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## biophysical_model_type   2 0.80397 0.40198  23.049 5.503e-09 ***
## Residuals              102 1.77891 0.01744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response s_RMSE :
##                          Df Sum Sq Mean Sq F value    Pr(>F)
## biophysical_model_type   2 3.4823 1.74114    80.9 < 2.2e-16 ***
## Residuals              102 2.1953 0.02152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response sev :
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## biophysical_model_type   2  587.43 293.714  24.804 1.667e-09 ***
## Residuals              102 1207.83  11.841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response sev.stdsub :
##                          Df  Sum Sq  Mean Sq F value    Pr(>F)
## biophysical_model_type   2 0.55043 0.275216  36.561 1.067e-12 ***
## Residuals              102 0.76781 0.007528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response TP.1 :
##                          Df  Sum Sq  Mean Sq F value   Pr(>F)
## biophysical_model_type   2 0.23934 0.119672  36.639 1.02e-12 ***
```

```
## Residuals              102 0.33315 0.003266
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response TP.25 :
##                          Df  Sum Sq Mean Sq F value    Pr(>F)
## biophysical_model_type   2 0.72504 0.36252  54.782 < 2.2e-16 ***
## Residuals              102 0.67499 0.00662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUC and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.29    -
## NMDA    2.9e-13 1.1e-12
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUC.1 and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.0014  -
## NMDA    1.1e-15 < 2e-16
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUCstd and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.54    -
## NMDA    1.7e-09 1.4e-08
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$D.prime and models$biophysical_model_type
##
##         AMPA   AMPA_SK
## AMPA_SK 0.012  -
## NMDA    <2e-16 <2e-16
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$s_MAE and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.26    -
## NMDA    2.1e-07 7.4e-08
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$s_RMSE and models$biophysical_model_type
##
##         AMPA   AMPA_SK
## AMPA_SK 0.042  -
## NMDA    <2e-16 <2e-16
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$sev and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.25    -
## NMDA    8.0e-08 2.9e-08
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$sev.stdsub and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.17    -
## NMDA    2.3e-10 6.9e-11
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$TP.1 and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.049   -
## NMDA    1.4e-09 1.4e-11
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$TP.25 and models$biophysical_model_type
##
##         AMPA    AMPA_SK
## AMPA_SK 0.0079  -
## NMDA    2.4e-12 1.5e-15
##
## P value adjustment method: none
```

```
##                Df  Pillai approx F num Df den Df    Pr(>F)
## NN_model_type   1 0.53987   11.029     10     94 3.262e-12 ***
## Residuals     103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##  Response AUC :
##                Df    Sum Sq    Mean Sq F value  Pr(>F)
## NN_model_type   1 0.0002825 2.8254e-04  3.8455 0.05258 .
## Residuals     103 0.0075678 7.3474e-05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response AUC.1 :
##                Df  Sum Sq  Mean Sq F value    Pr(>F)
## NN_model_type   1 0.17902 0.179021  12.951 0.0004932 ***
## Residuals     103 1.42370 0.013822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response AUCstd :
##                Df     Sum Sq    Mean Sq F value Pr(>F)
## NN_model_type   1 1.2710e-07 1.2711e-07  0.8956 0.3462
## Residuals     103 1.4618e-05 1.4192e-07
##
##  Response D.prime :
##                Df  Sum Sq Mean Sq F value  Pr(>F)
## NN_model_type   1  0.7815 0.78150  6.8702 0.01009 *
## Residuals     103 11.7164 0.11375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response s_MAE :
##                Df  Sum Sq   Mean Sq F value Pr(>F)
## NN_model_type   1 0.00431 0.0043062   0.172 0.6792
## Residuals     103 2.57857 0.0250346
##
##  Response s_RMSE :
##                Df Sum Sq  Mean Sq F value Pr(>F)
## NN_model_type   1 0.0003 0.000256  0.0046 0.9458
## Residuals     103 5.6773 0.055119
##
##  Response sev :
##                Df  Sum Sq Mean Sq F value Pr(>F)
## NN_model_type   1    1.23  1.2299  0.0706  0.791
## Residuals     103 1794.03 17.4177
##
##  Response sev.stdsub :
##                Df  Sum Sq   Mean Sq F value Pr(>F)
## NN_model_type   1 0.00041 0.0004083  0.0319 0.8586
## Residuals     103 1.31784 0.0127945
##
##  Response TP.1 :
##                Df  Sum Sq  Mean Sq F value    Pr(>F)
## NN_model_type   1 0.09568 0.095682  20.669 1.493e-05 ***
## Residuals     103 0.47682 0.004629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response TP.25 :
##                Df  Sum Sq  Mean Sq F value    Pr(>F)
## NN_model_type   1 0.19684 0.196836   16.85 8.115e-05 ***
## Residuals     103 1.20320 0.011682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUC and models$NN_model_type
##
##     FCN
## TCN 0.053
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUC.1 and models$NN_model_type
##
##     FCN
## TCN 0.00049
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUCstd and models$NN_model_type
##
##     FCN
## TCN 0.35
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$D.prime and models$NN_model_type
##
##     FCN
## TCN 0.01
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$s_MAE and models$NN_model_type
##
##     FCN
## TCN 0.68
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$s_RMSE and models$NN_model_type
##
##     FCN
## TCN 0.95
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$sev and models$NN_model_type
##
##     FCN
## TCN 0.79
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$sev.stdsub and models$NN_model_type
##
##     FCN
## TCN 0.86
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$TP.1 and models$NN_model_type
##
##     FCN
## TCN 1.5e-05
##
## P value adjustment method: none
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$TP.25 and models$NN_model_type
##
##     FCN
## TCN 8.1e-05
##
## P value adjustment method: none
```

```
## [1] 0.005
```

After manova

Perform a MANOVA to test whether any of your numeric variables (or a subset of them, if including them all doesn't make sense) show a mean difference across levels of one of your categorical variables. #### If significant, perform univariate ANOVAs to find response(s) showing a mean difference across groups, and perform post-hoc t tests to find which groups differ. Discuss the number of tests you have performed, calculate the probability of at least one type I error, and adjust the significance level accordingly (Bonferroni correction) before discussing significant differences. Briefly discuss assumptions and whether or not they are likely to have been met.

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## NN_model_type   1 0.0957 0.09568   20.67 1.49e-05 ***
## Residuals     103 0.4768 0.00463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$TP.1 and models$NN_model_type
##
##     FCN
## TCN 1.5e-05
##
## P value adjustment method: none
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## NN_model_type   1 0.1968 0.19684   16.85 8.12e-05 ***
## Residuals     103 1.2032 0.01168
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$TP.25 and models$NN_model_type
##
##     FCN
## TCN 8.1e-05
##
## P value adjustment method: none
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## NN_model_type   1  0.179 0.17902   12.95 0.000493 ***
## Residuals     103  1.424 0.01382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  models$AUC.1 and models$NN_model_type
##
##     FCN
## TCN 0.00049
##
## P value adjustment method: none
```

one type 1 error Discuss the number of tests you have performed, calculate the probability of at least one type I error, and adjust the significance level accordingly (Bonferroni correction) before discussing significant differences.

```
# probability that you have made at least one type I error
prob <- 1 - (.95^10)
prob
```

```
## [1] 0.4012631
```

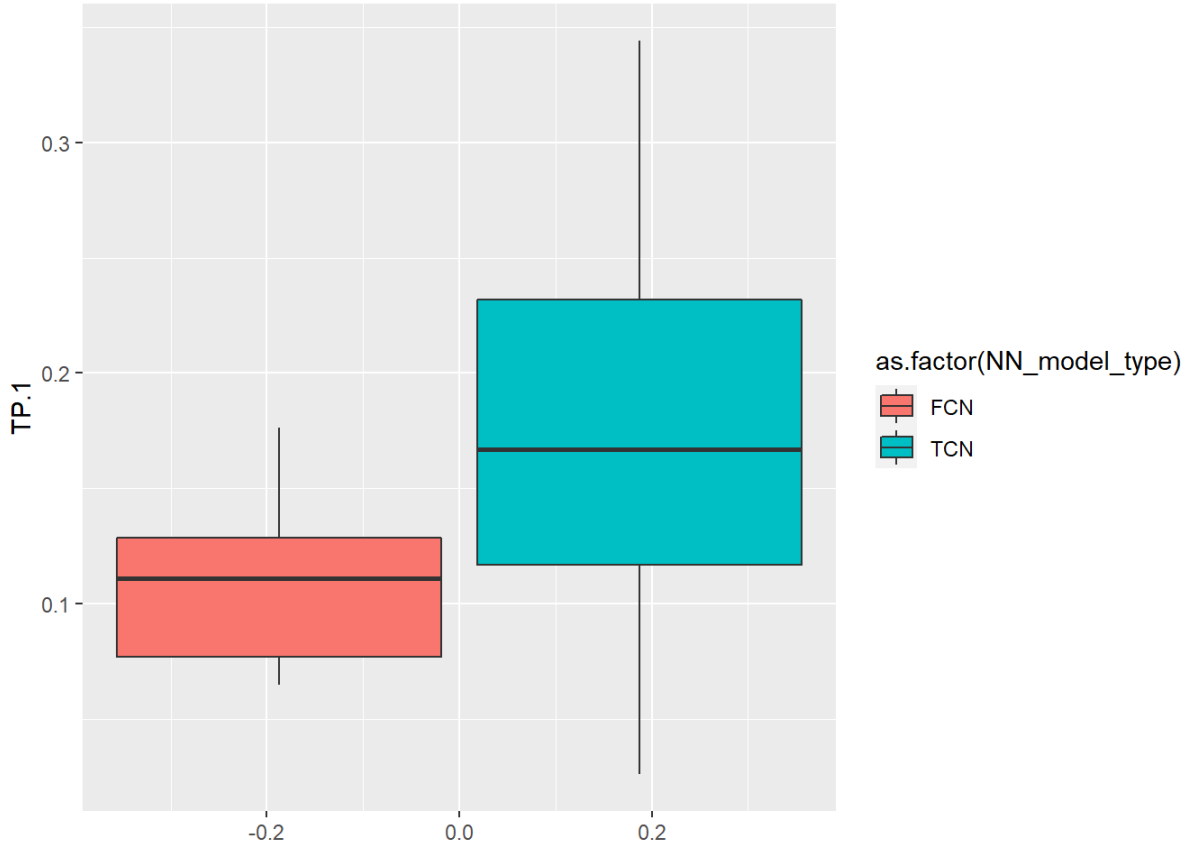
```
# Bonferroni adjusted
bon_adj <- .05/10
bon_adj
```

```
## [1] 0.005
```

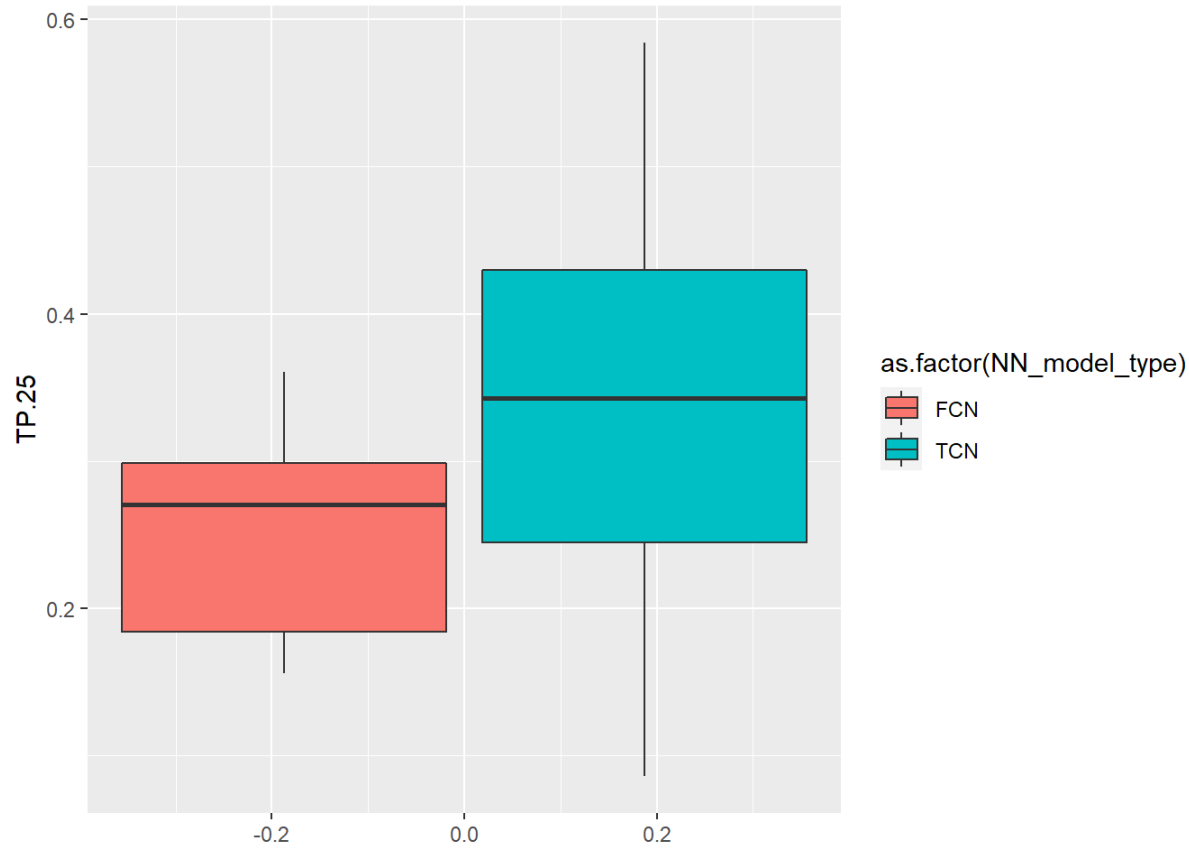
similar significances

Briefly discuss assumptions and whether or not they are likely to have been met.

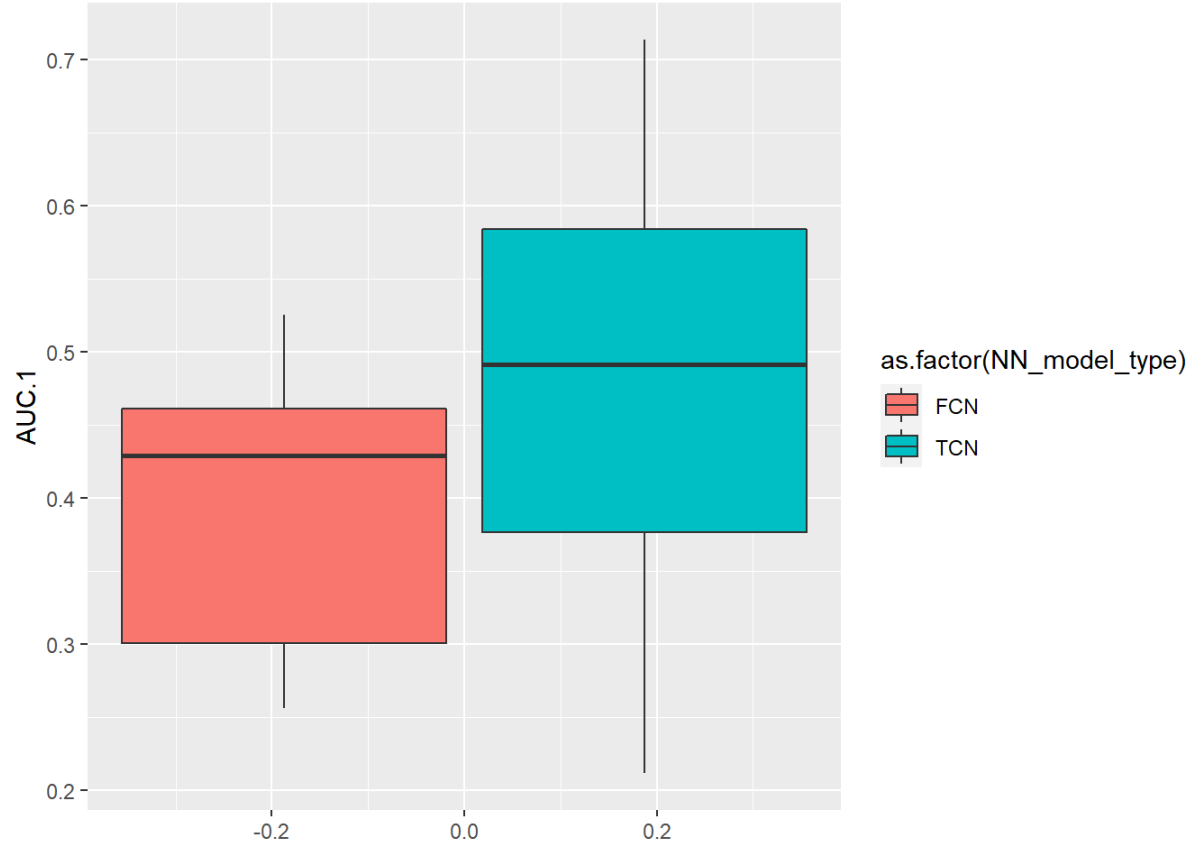
```
# Check assumptions visually
ggplot(models, aes(y = TP.1)) +
  geom_boxplot(aes(fill = as.factor(NN_model_type)))
```




```
# Check assumptions visually
ggplot(models, aes(y = TP.25)) +
  geom_boxplot(aes(fill = as.factor(NN_model_type)))
```




```
# Check assumptions visually
ggplot(models, aes(y = AUC.1)) +
  geom_boxplot(aes(fill = as.factor(NN_model_type)))
```



Randomization Test

Perform a randomization test on your data (that makes sense and is interesting to look at). This can be anything you want! State the null and alternative hypotheses, perform the test, and interpret the results. Create a plot visualizing the null distribution and the test statistic.

null hypothesis: The null hypothesis is that the observed patten is no different than what we would expect by random chance.
altnerative hyptohesis: The alternative hypothesis is that the observed patten is different than what we would expect by random chance.

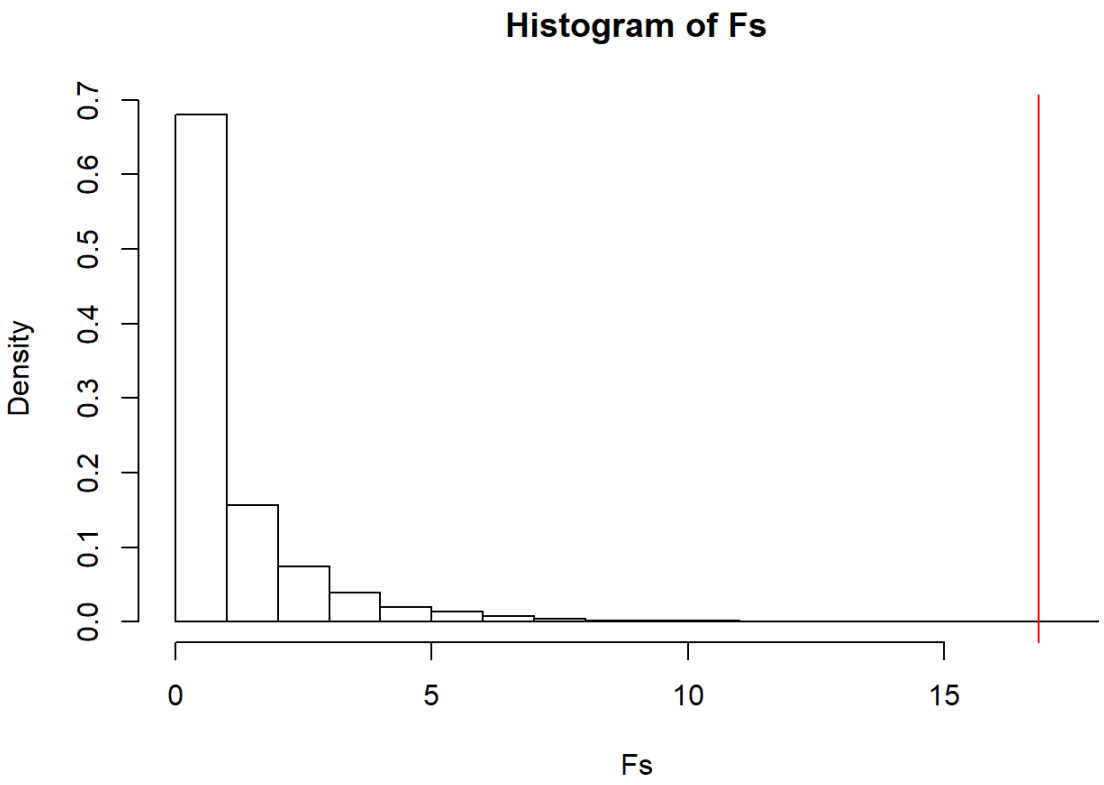
```
# do randomization test on TP.25
# Observed F-statistic, running anova
obs_F <- 16.85

# find dimensions of dataset to determine MSB and MSW later on
dim(models)
```

```
## [1] 105  18
```

```
# Randomization test (using replicate)
Fs <- replicate(5000,{
  # Randomly permute the response variable across doses
  new <- models %>%
    mutate(TP.25= sample(TP.25))
  # Compute variation within groups
  SSW <- new %>%
    group_by(NN_model_type) %>%
    summarize(SSW = sum((TP.25 - mean(TP.25))^2)) %>%
    summarize(sum(SSW)) %>%
    pull
  # Compute variation between groups
  SSB <- new %>%
    mutate(mean = mean(TP.25)) %>%
    group_by(NN_model_type) %>%
    mutate(groupmean = mean(TP.25)) %>%
    summarize(SSB = sum((mean - groupmean)^2)) %>%
    summarize(sum(SSB)) %>%
    pull
  # Compute the F-statistic (ratio of MSB and MSW)
  # df for SSB is 3 groups - 1 = 2
  # df for SSW is 105 observations - 2 groups = 103
  (SSB/1)/(SSW/103)
})

# Calculate the proportion of F statistic that are greater than the observed F-statistic
mean(Fs > obs_F)
```

```
## [1] 2e-04
```

```
# Represent the distribution of the F-statistics for each randomized sample
hist(Fs, prob=T); abline(v = obs_F, col="red",add=T)
```



Results: Yes,

statistically significant Goal: scramble data to break any associations present within or between the data. on average, themeans will be the same across the groups when doing this. do the scramble amny times and record f stat each time. Then compare distribution of many f stat from randomization to the f stat of the sample. If the f stat of the sample is significant, it will be far away on the graph and it is.

Linear Regression model

Build a linear regression model predicting one of your numeric response variables from at least 2 explanatory variables, including their interaction. #### □ Mean-center any numeric variables involved in the interaction. □ Create a graph to visualize the interaction between 2 variables on the response. □ Interpret the coefficient estimates in context (regardless of significance). □ What proportion of the variation in the response does your model explain? □ Check assumptions of linearity, normality, and homoscedasticity either graphically or using a hypothesis test (or both). □ Regardless of meeting the assumptions, recompute regression results with robust standard errors. Discuss significance of results, including any changes from before/after calculating robust SEs if applicable. □ Finally, compute bootstrapped standard errors. Discuss any changes you obser

```
# Include an interaction term in the regression model
#fit <- lm(AUC.1 ~ biophysical_model_type * D.prime, data = models)
#summary(fit)

# Center the data around the means (the intercept becomes more informative)
models$D_c <- models$D.prime - mean(models$D.prime)

# Include an interaction term in the regression model with centered predictors
fit_c <- lm(AUC.1 ~ biophysical_model_type * D_c, data = models)
summary(fit_c)
```

```
##
## Call:
## lm(formula = AUC.1 ~ biophysical_model_type * D_c, data = models)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.064252 -0.014303 -0.002303  0.016282  0.095103
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       0.426559   0.006380  66.858  < 2e-16 ***
## biophysical_model_typeAMPA_SK    -0.008124   0.014133  -0.575    0.567
## biophysical_model_typeNMDA       -0.011359   0.009198  -1.235    0.220
## D_c                               0.492123   0.027146  18.129  < 2e-16 ***
## biophysical_model_typeAMPA_SK:D_c  0.016110   0.044455   0.362    0.718
## biophysical_model_typeNMDA:D_c    -0.276811   0.031461  -8.798 4.54e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02779 on 99 degrees of freedom
## Multiple R-squared:  0.9523, Adjusted R-squared:  0.9499
## F-statistic: 395.2 on 5 and 99 DF,  p-value: < 2.2e-16
```

Interpretations

The biophysical model type AMPA_SK is NOT significantly associated with AUC.1 for the biophysical model type, AMPA: for every one unit increase in AMPA_SK, the AUC.1 value goes down by 0.008 (t = -0.575, df = 99, p = 0.567).

The biophysical model type NMDA is NOT significantly associated with AUC.1 for the biophysical model type, AMPA: for every one unit increase in NMDA, the AUC.1 value goes down by 0.011 (t = -1.235, df = 99, p = 0.220).

D.prime is significantly associated with AUC.1 for the biophysical model type, AMPA: for every one unit increase in D.prime, the AUC.1 value goes up by 0.4921 (t = 18.129, df = 99, p < 0.001).

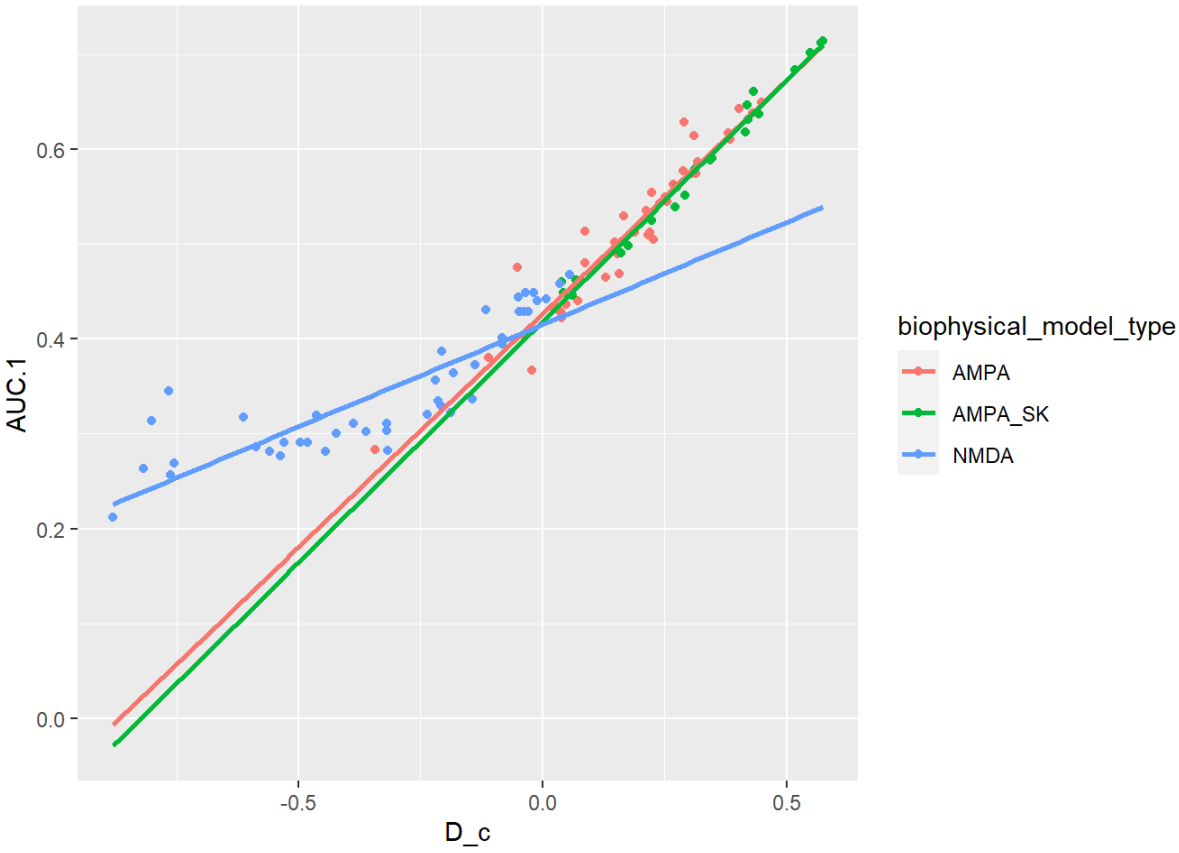
There is NOT a significant interaction between the the AMPA_SK biophysical model type and D.prime. The slope for D.prime on AUC.1 is 0.01611 higher for the AMPA_SK biophysical model type compared to the AMPA biophysical model type (t = 0.362, df= 99, p = 0.718).

There is a significant interaction between the the NMDA biophysical model type and D.prime. The slope for D.prime on AUC.1 is 0.2768 lower for the NMDA biophysical model type compared to the AMPA biophysical model type (t = -8.798, df= 99, p < 0.001).

Create a graph to visualize the interaction between 2 variables on the response.

```
# Create a graph to visualize the interaction between D.prime and AUC.1 on the biophysical model type
ggplot(models, aes(x = D_c, y = AUC.1, color = biophysical_model_type)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE, fullrange=TRUE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



What proportion of the variation in the response does your model explain?

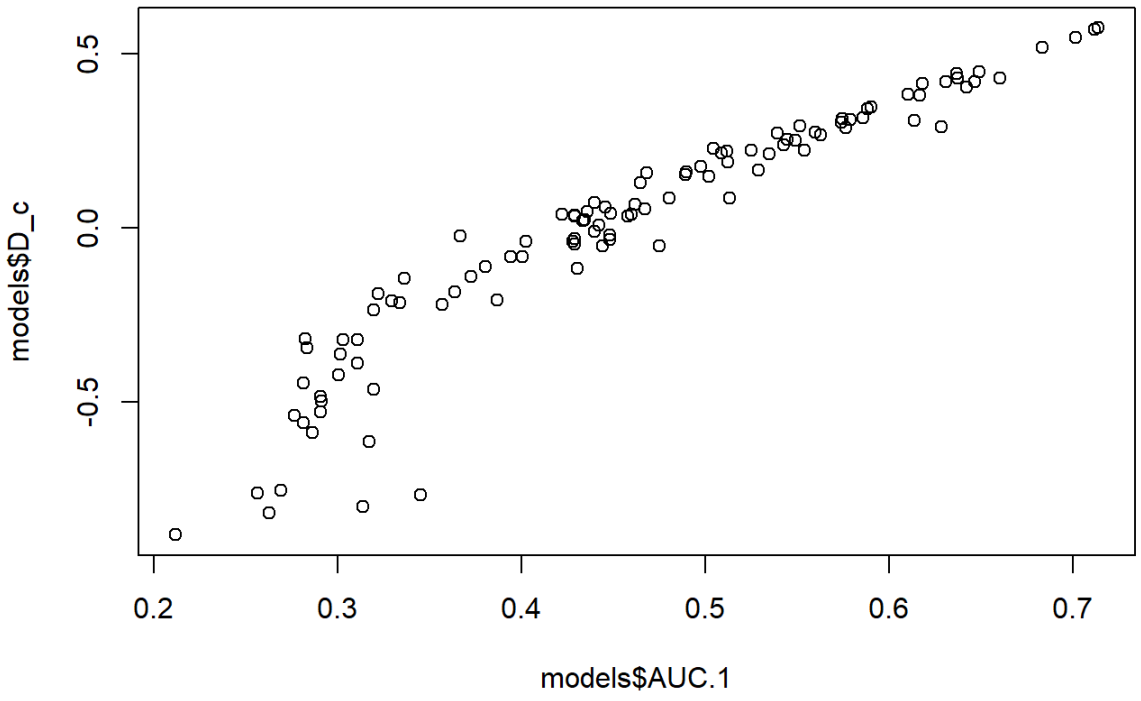
```
# calculate r-squared value using built-in function
summary(fit_c)$r.sq
```

```
## [1] 0.9522938
```

According to the mean-centered distribution, 95.23% of the variation is explained by the model.

Check assumptions of linearity, normality, and homoscedasticity either graphically or using a hypothesis test (or both).

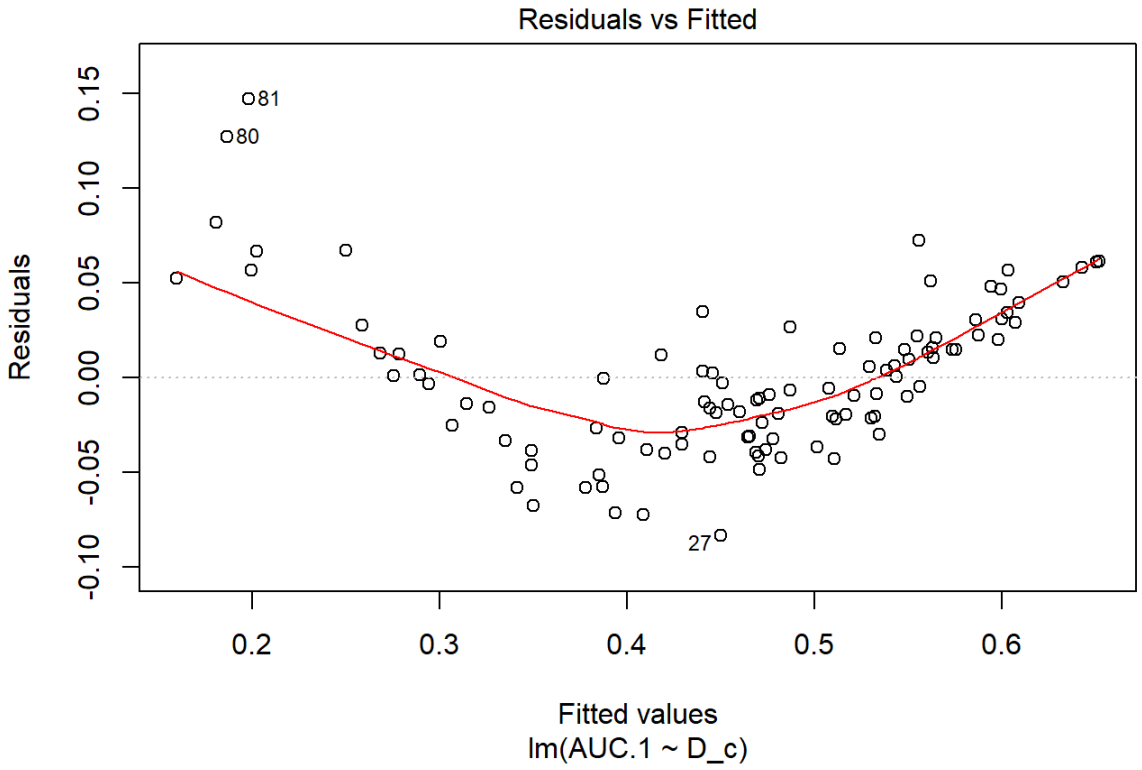
```
# check assumptions visually
# Recall the example of predicting distance to stop by speed of a car
plot(models$AUC.1, models$D_c)
```



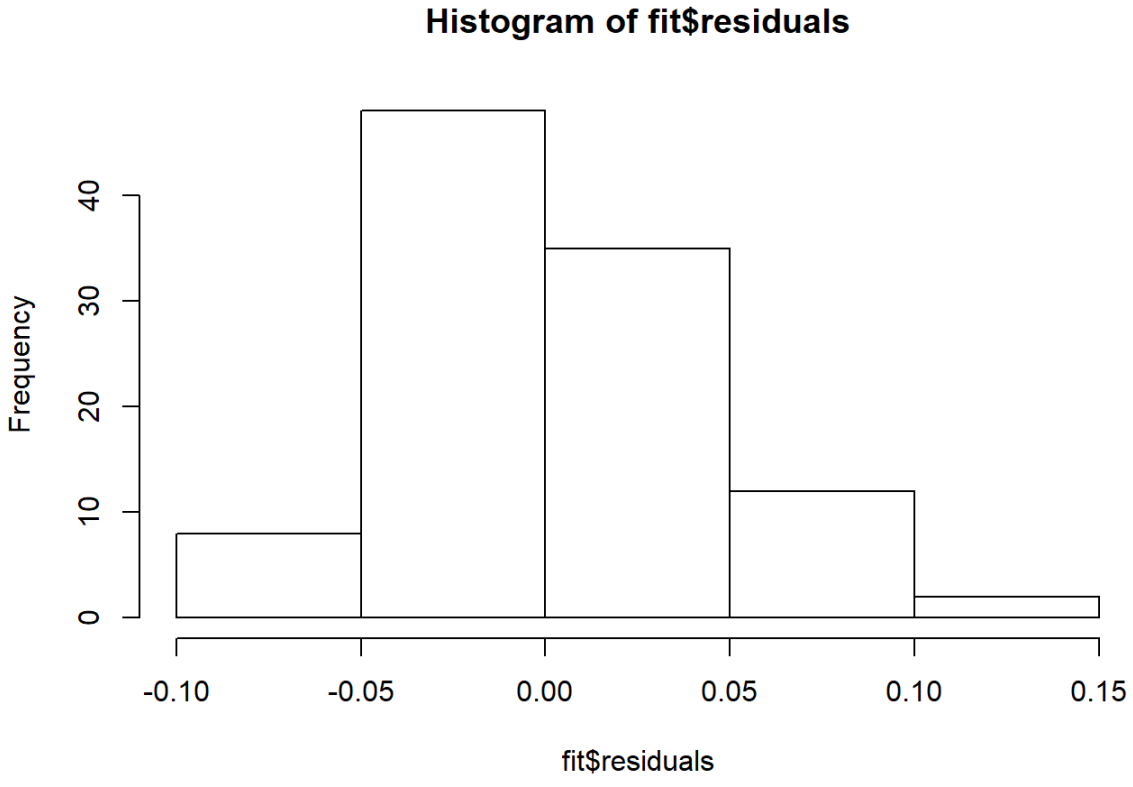

```
fit <- lm(AUC.1 ~ D_c, data = models)
summary(fit)
```

```
##
## Call:
## lm(formula = AUC.1 ~ D_c, data = models)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.083442 -0.030062 -0.004895  0.021197  0.147100
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.457582   0.003981  114.93   <2e-16 ***
## D_c         0.338411   0.011540   29.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0408 on 103 degrees of freedom
## Multiple R-squared:  0.893,  Adjusted R-squared:  0.892
## F-statistic: 859.9 on 1 and 103 DF,  p-value: < 2.2e-16
```

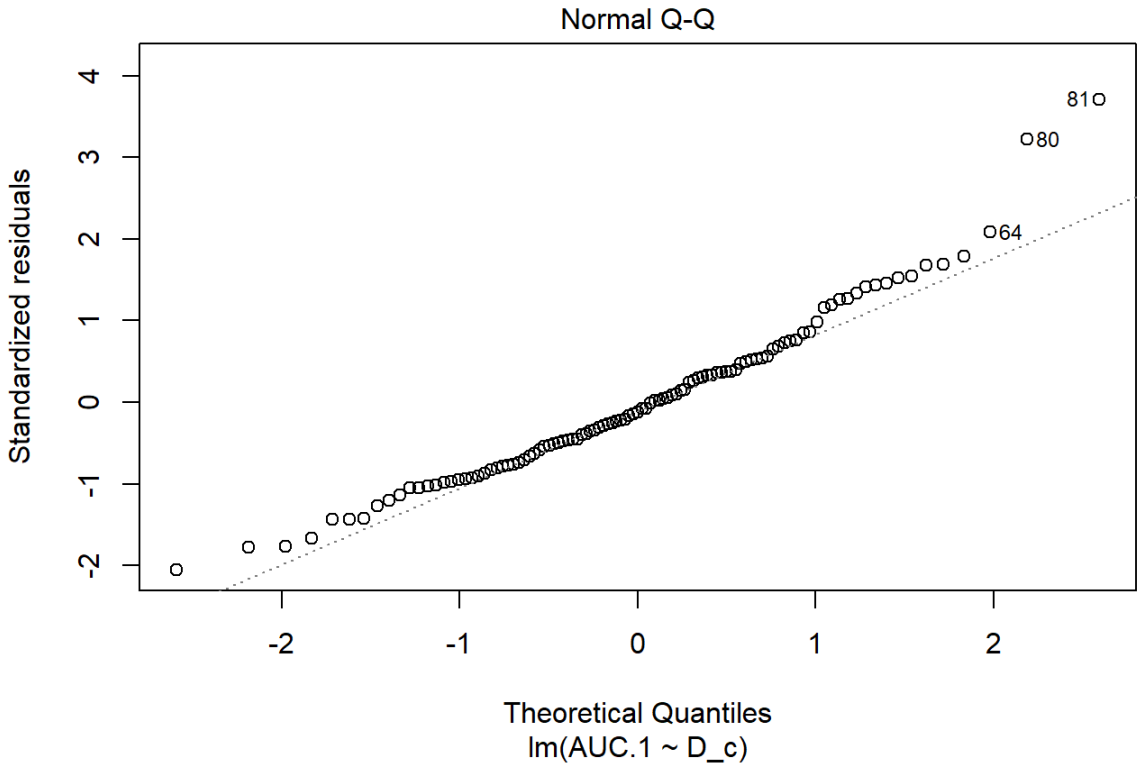
```
# Residuals vs Fitted values plot
plot(fit, which = 1)
```




```
# Histogram of residuals
hist(fit$residuals)
```



```
# Q-Q plot for the residuals
plot(fit, which = 2)
```




```
# check assumptions numerically: normality
# Shapiro-Wilk test
# H0: normality
shapiro.test(fit$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.96708, p-value = 0.01031
```

```
# Kolmogorov-Smirnov test
# H0: normality
ks.test(fit$residuals, "pnorm", mean=0, sd(fit$residuals))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  fit$residuals
## D = 0.061811, p-value = 0.8173
## alternative hypothesis: two-sided
```

```
  # note: the error indicates that there are repeated values for the residuals

# Check assumptions numverically: homoscedasticity
library(sandwich);
# Install a new package
# install.packages("lmtest")
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Breusch-Pagan test
# H0: homoscedasticity
bptest(fit)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit
## BP = 13.869, df = 1, p-value = 0.000196
```

Passes asumptions. While visualizations look like the data might not pass tests, the numerical assumption tests produced very low p values on the one sample tests suggesting a normal distribution.

Regardless of meeting the assumptions, recompute regression results with robust standard errors. Discuss significance of results, including any changes from before/after calculating robust SEs if applicable.

```
# Robust Standard Errors
# install.packages("sandwich")
library(sandwich)
coeftest(fit, vcov = vcovHC(fit))
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 0.4575816  0.0040735 112.332 < 2.2e-16 ***
## D_c         0.3384114  0.0182066  18.587 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# original values
# 0.492123   0.027146  18.129  < 2e-16 ***
```

There was no significant diffference before and after calculating robust SEs. The results are still statisitcally siginifcant this time with: D.prime is significantly associated with AUC.1 for the biophysical model type, AMPA: for every one unit increase in D.prime, the AUC.1 value goes up by 0.338 ($t = 18.587$, $df = 99$, $p < 0.001$).

Finally, compute bootstrapped standard errors. Discuss any changes you observe in SEs and p-values using these SEs compared to the original SEs and the robust SEs

```
# When assumptions are violated (homoscedasticity, normality, small sample size)
# use bootstrap samples to estimate coefficients, SEs, fitted values, ...

# Example of estimating coefficients SEs
# Use the function replicate to repeat the process (similar to a for loop)
samp_SEs <- replicate(5000, {
  # Bootstrap your data (resample observations)
  boot_data <- sample_frac(models, replace = TRUE)
  # Fit regression model
  fitboot <- lm(AUC.1 ~ D_c, data = boot_data)
  # Save the coefficients
  coef(fitboot)
})

# Estimated SEs
samp_SEs %>%
  # Transpose the obtained matrices
  t %>%
  # Consider the matrix as a data frame
  as.data.frame %>%
  # Compute the standard error (standard deviation of the sampling distribution)
  summarize_all(sd)
```

```
##   (Intercept)        D_c
## 1 0.003923776 0.01768234
```

```
# We can also consider a confidence interval for the estimates
samp_SEs %>%
  # Transpose the obtained matrices
  t %>%
  # Consider the matrix as a data frame
  as.data.frame %>%
  # Pivot longer to group by and summarize each coefficient
  pivot_longer(everything(), names_to = "estimates", values_to = "value") %>%
  group_by(estimates) %>%
  summarize(lower = quantile(value,.025), upper = quantile(value,.975))
```

```
## # A tibble: 2 x 3
##   estimates   lower upper
## * <chr>       <dbl> <dbl>
## 1 (Intercept) 0.450 0.465
## 2 D_c         0.306 0.377
```

```
# Compare to original fit
confint(fit, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.4496853 0.4654779
## D_c         0.3155238 0.3612990
```

Logistic Regression

Build a logistic regression model predicting a binary categorical variable (if you don’t have one, create one based on another variable or combination of other variables) from at least 2 explanatory variables (interaction is not necessary). Interpret coefficient estimates in context (regardless of significance). Report a confusion matrix for your logistic regression. Compute and discuss the Accuracy, Sensitivity (TPR), Specificity (TNR), and Recall (PPV). Create a graph to plot density of log-odds (logit) by your binary outcome variable. Generate a ROC curve (plot) and calculate AUC (either manually or with a package) and interpret the results. #### Interpret coefficient estimates in context (regardless of significance).

```
# binary categorical variable is NN_model_type
# Create a binary variable coded as 0 and 1
models <- models %>%
  mutate(y = ifelse(NN_model_type == "FCN", 1, 0))

# Consider a logistic model with the two numeric variables, TP.25  and s_RMSE
log_model <- glm(y ~ TP.25 + s_RMSE, data = models, family = "binomial")
summary(log_model)
```

```
##
## Call:
## glm(formula = y ~ TP.25 + s_RMSE, family = "binomial", data = models)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3496  -0.3601  -0.0899   0.2794   1.6325
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   33.716      7.340   4.593 4.36e-06 ***
## TP.25        -59.115     12.657  -4.670 3.01e-06 ***
## s_RMSE       -25.371      5.596  -4.534 5.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 127.423  on 104  degrees of freedom
## Residual deviance:  57.207  on 102  degrees of freedom
## AIC: 63.207
##
## Number of Fisher Scoring iterations: 7
```

interpretations

Interpretations for coefficient of TP.25 holding s_RMSE constant: a one unit increase in TP.25 decreases the log-odds of the NN model type being FCN by 59.115.

Interpretations for coefficient of s_RMSE holding TP.25 constant: a one unit increase in s_RMSE decreases the log-odds of the NN model type being FCN by 25.371.

Report a confusion matrix for your logistic regression.

```
# outcome = NN_model_type
# Add predicted probabilities to the dataset
models$prob <- predict(log_model, type = "response")

# Predicted outcome is based on the probability of malignant
# if the probability is greater than 0.5, the NN model type is FCN
models$predicted <- ifelse(models$prob > .5, "FCN", "TCN")
# Confusion matrix
table(truth = models$NN_model_type, prediction = models$predicted)
```

```
##      prediction
## truth FCN TCN
##   FCN  23   8
##   TCN   4  70
```

```
# Accuracy (correctly classified cases)
(23 + 70)/105
```

```
## [1] 0.8857143
```

```
# Sensitivity (True Positive Rate, TPR)
70/74
```

```
## [1] 0.9459459
```

```
# Specificity (True Negative Rate, TNR)
23/31
```

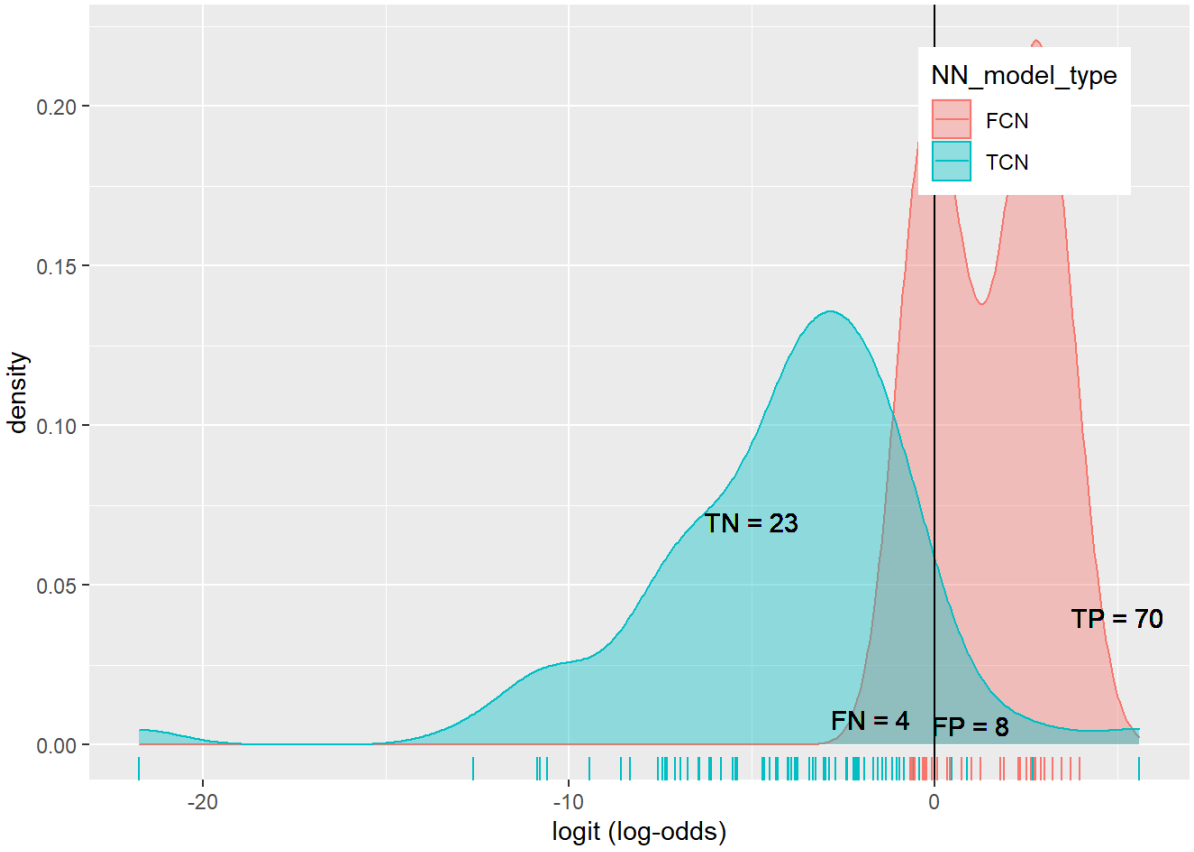
```
## [1] 0.7419355
```

```
# Precision (Positive Predictive Value, PPV)
70/78
```

```
## [1] 0.8974359
```

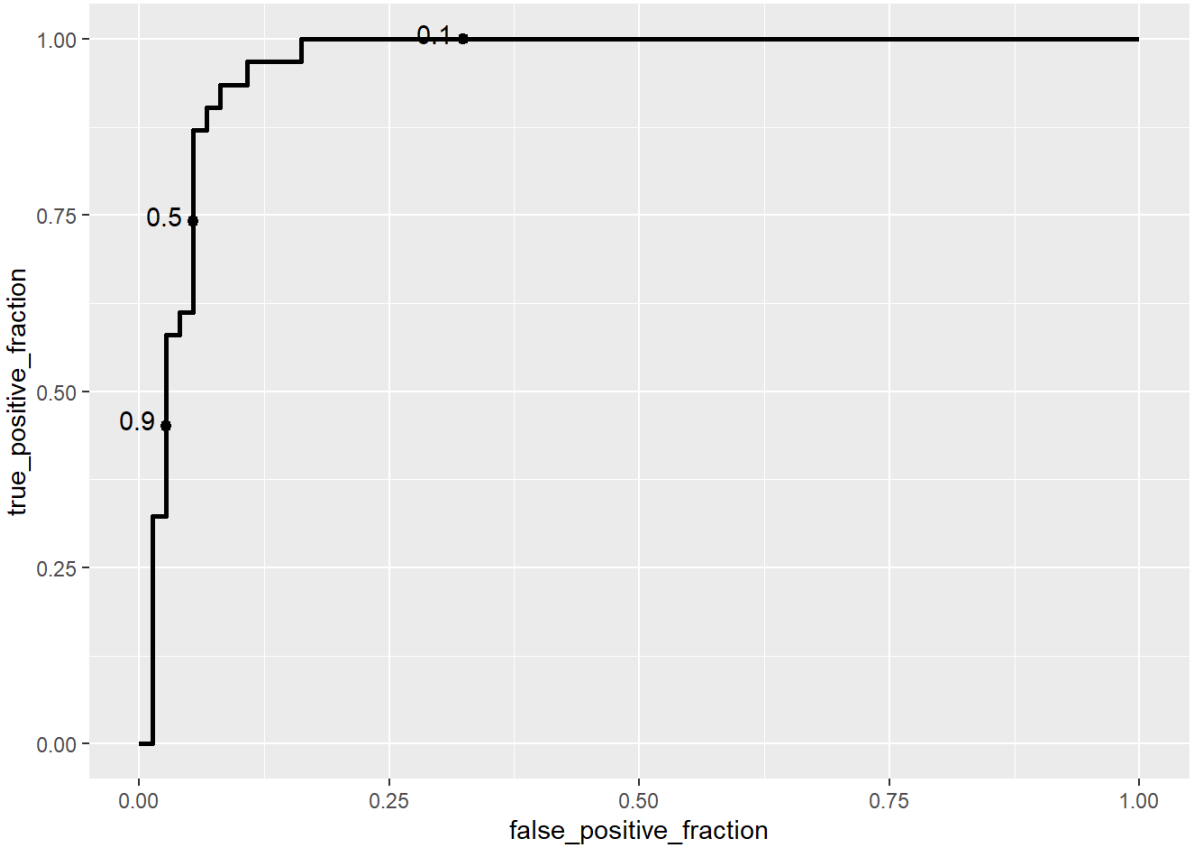
Create a graph to plot density of log-odds (logit) by your binary outcome variable.

```
# binary categorical variable is NN_model_type

# Predicted log odds
models$logit <- predict(log_model, type = "link")

# Density plot of log-odds for each outcome
models %>%
  ggplot() +
  geom_density(aes(logit, color = NN_model_type, fill = NN_model_type), alpha = .4) +
    geom_rug(aes(logit, color = NN_model_type)) +
  geom_text(x = -5, y = .07, label = "TN = 23") +
  geom_text(x = -1.75, y = .008, label = "FN = 4") +
  geom_text(x = 1, y = .006, label = "FP = 8") +
  geom_text(x = 5, y = .04, label = "TP = 70") +
  theme(legend.position = c(.85,.85)) +
  geom_vline(xintercept = 0) +
  xlab("logit (log-odds)")
```



Generate a ROC curve (plot) and calculate AUC (either manually or with a package) and interpret the results

```
# Call the library plotROC
library(plotROC)

# Plot ROC depending on values of y and its probabilities displaying some cutoff values
ROCplot1 <- ggplot(models) +
  geom_roc(aes(d = y, m = prob), cutoffs.at = list(0.1, 0.5, 0.9))
ROCplot1
```




```
# Calculate the area under the curve still using the library plotROC with function calc_auc
calc_auc(ROCplot1)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9598954
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.