# Evaluating logistic regression

Stephanie J. Spielman

Data Science for Biologists, Spring 2020

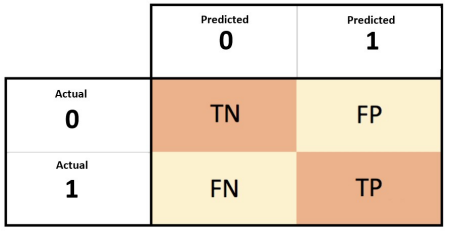
# Some notes

- Stepwise model selection is *not a rule*
- In linear regression, one often uses $R^2$ and RMSE values to compare different viable models

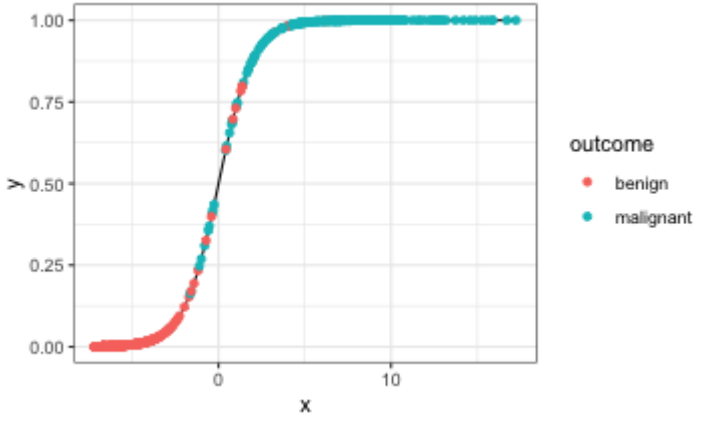
# Some notes

- Stepwise model selection is *not a rule*
- In linear regression, one often uses $R^2$ and RMSE values to compare different viable models
- In *logistic regression*, we use confusion matrix calculations and **AUC** (area under the curve… what curve?)

- $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*
- $FPR = FP/N = \frac{FP}{FP+TN}$
  - AKA *1 - specificity*
- Precision: $PPV = \frac{TP}{TP+FP}$
  - AKA *positive predictive value*
- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$



```
biopsy <- read_csv(paste0("https://raw.githubusercontent.com/sjspielman/",
                          "datascience_for_biologists/master/slides/biopsy.csv"))

## Build the model
biopsy %>%
  mutate(outcome = case_when(outcome == "malignant" ~ 1,  ## "success" in model
                             outcome == "benign" ~ 0)) -> biopsy_fct
baseline_logit_fit <- glm( outcome ~ ., data = biopsy_fct, family = "binomial")
selected_fit       <- step(baseline_logit_fit, trace = F)

## Extract the model
tibble(x = selected_fit$linear.predictors,
       y = selected_fit$fitted.values,
       outcome = biopsy$outcome) -> extracted_model

## Plot the model
extracted_model %>%
  ggplot(aes(x = x, y = y)) +
    geom_line() +
    geom_point(aes(color = outcome))
```



# Predictions with logistic regressions

```
broom::tidy(selected_fit) %>%
  dplyr::select(term)
## # A tibble: 8 x 1
##   term
##   <chr>
## 1 (Intercept)
## 2 clump_thickness
## 3 uniform_cell_shape
## 4 marg_adhesion
## 5 bare_nuclei
## 6 bland_chromatin
## 7 normal_nucleoli
## 8 mitoses

## Predict! The new data is in a tibble
tibble(clump_thickness    = 3,
       uniform_cell_shape = 2,
       marg_adhesion      = 2,
       bare_nuclei        = 1,
       bland_chromatin    = 4,
       normal_nucleoli    = 2,
       mitoses            = 3) -> new_biopsy
```

# Predictions with logistic regressions

```
predict(selected_fit, new_biopsy)
##         1
## -2.723465
```

# Predictions with logistic regressions

```
predict(selected_fit, new_biopsy)
##         1
## -2.723465
```

```
predict(selected_fit, new_biopsy, type = "response")
##          1
## 0.06160284
```

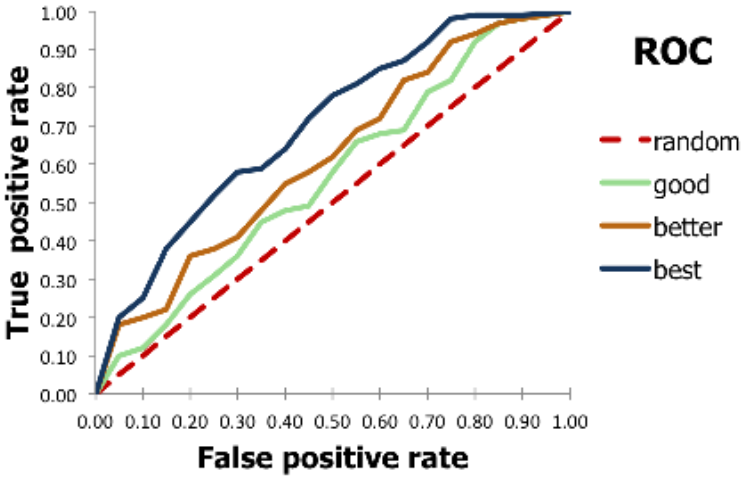
```
t <- -2.723465
1 / (1 + exp(-1*t))
## [1] 0.06160286
```

# Evaluating logistic regressions

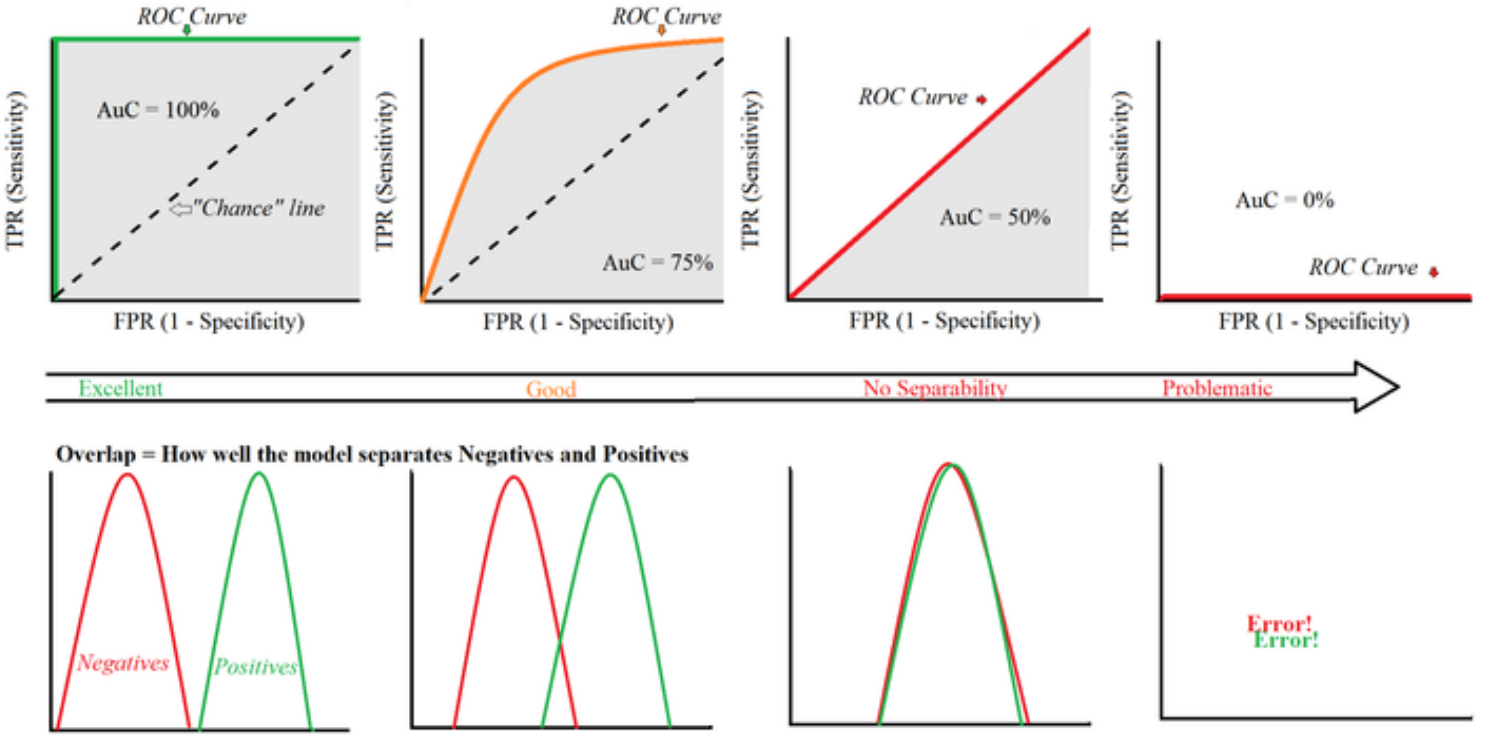
**Receiver Operating Characteristic** Curve

- TPR on Y-axis
- FPR (1 - specificity) on X-axis
- The AUC (**a**rea **u**nder the **c**urve) is an overall assessment of performance *at any threshold*

- $TPR = TP/P = \frac{TP}{TP+FN}$ (*sensitivity* AKA *recall*)
- $TNR = TN/N = \frac{TN}{FP+TN}$ (*specificity*)
- $FPR = FP/N = \frac{FP}{FP+TN}$ (*1 - specificity*)

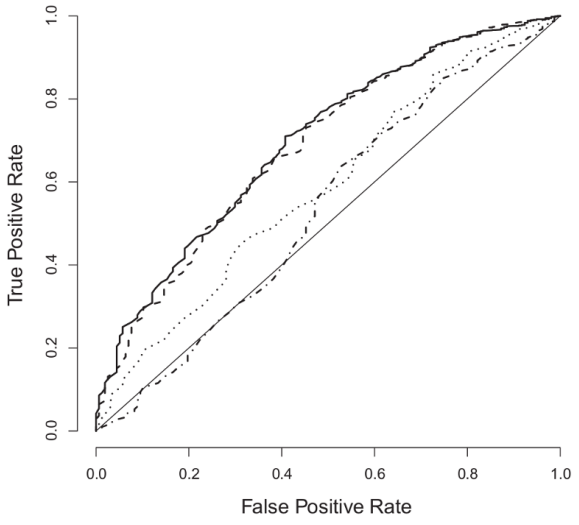


# Getting a "feel" for ROC curves

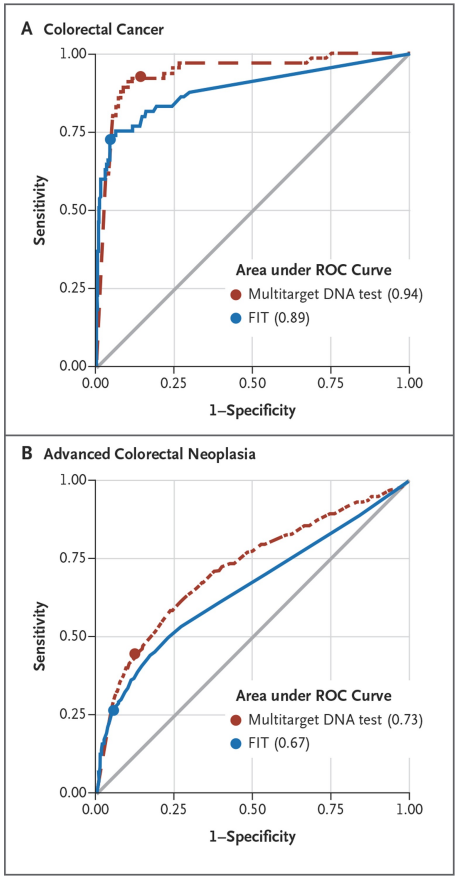


# Examples of ROC curves in the literature

Keller et al. Genome Biol Evol 2012; 4:80-88




Imperiale et al. N Engl J Med 2014; 370:1287-1297



# ROC vs PR

- ROC curves are suitable when data is *balanced*
  - Similar amounts of positives, negatives in the dataset
  - FPR (1 - specificity) on X-axis, TPR on Y-axis
- **Precision-Recall** curves are more suitable for *unbalanced* data
  - Precision (PPV) on Y-axis, recall (TPR) on X-axis

- $TPR = TP/P = \frac{TP}{TP+FN}$ (*recall*)
- $FPR = FP/N = \frac{FP}{FP+TN}$
- $PPV = \frac{TP}{TP+FP}$

# Is the biopsy data balanced?

```
biopsy %>%
  count(outcome)
## # A tibble: 2 x 2
##   outcome       n
##   <chr>     <int>
## 1 benign      444
## 2 malignant   239
```

- About 2:1::benign:malignant
- Not very balanced, but it's reasonable. ROC is ok to use!
- *Problematically imbalanced* would be 4000 benign and 5 malignant (or vice versa).

# Making ROC curves

- Recall:
  - Our model fit is saved in **`selected_fit`**
  - Our data is saved in **`biopsy`**, but the model was built with **`biopsy_fct`**!!

```
## Installed for you in the cloud, but you need to install locally
#install.packages("pROC")
library(pROC)
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
# Use the function roc()
model_roc <- roc(biopsy_fct$outcome, selected_fit$linear.predictors)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

# This also works the same:
model_roc <- roc(biopsy_fct$outcome, selected_fit$fitted.values)
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

# Getting information out

```
model_roc$auc
## Area under the curve: 0.9963
```

- Models are usually *not this good*. This dataset comes from a package that teaches modeling - it was chosen for a reason..

# Getting information out

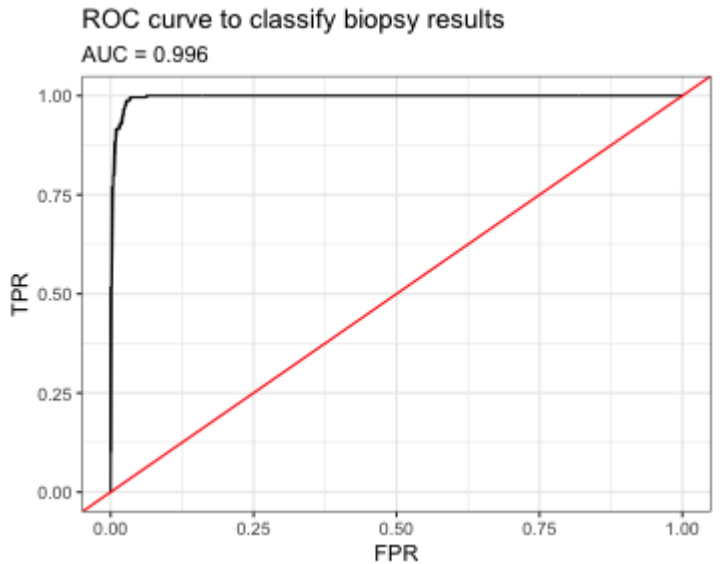
```
model_roc$auc
## Area under the curve: 0.9963
```

- Models are usually *not this good*. This dataset comes from a package that teaches modeling - it was chosen for a reason..

```
## Piped into head() to fit on the slide

## True positive rates
model_roc$sensitivities %>% head()
## [1] 1 1 1 1 1 1

## True negative rates
model_roc$specificities %>% head()
## [1] 0.00000000 0.07432432 0.07657658 0.08108108 0.08333333 0.08558559

## False positives rates
1 - model_roc$specificities %>% head()
## [1] 1.0000000 0.9256757 0.9234234 0.9189189 0.9166667 0.9144144
```

# Make an ROC curve

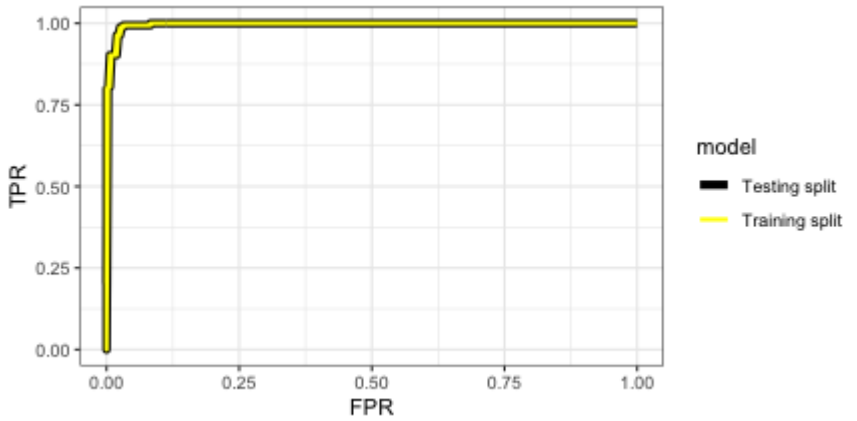
```
tibble(TPR = model_roc$sensitivities,
       FPR = 1 - model_roc$specificities) %>%
  ggplot(aes(x = FPR, y = TPR)) +
  geom_line() +
  labs(title = "ROC curve to classify biopsy results",
       subtitle = paste("AUC =",round(model_roc$auc, 3)) ) +
  ## this is the y=x line to GUIDE US!!
  geom_abline(col = "red")
```



# Comparing train, test splits

```
set.seed(1011)
biopsy %>%
  mutate(outcome = case_when(outcome == "malignant" ~ 1,  ## "success" in model
                             outcome == "benign" ~ 0)) -> biopsy_fct
baseline_logit_fit <- glm( outcome ~ ., data = biopsy_fct, family = "binomial")
selected_fit       <- step(baseline_logit_fit, trace = F)

## Training split and testing split
training_frac <- 0.7
biopsy_train  <- sample_frac(biopsy_fct, training_frac)
biopsy_test   <- anti_join(biopsy_fct, biopsy_train)
## Joining, by = c("clump_thickness", "uniform_cell_size", "uniform_cell_shape",
"marg_adhesion", "epithelial_cell_size", "bare_nuclei", "bland_chromatin",
"normal_nucleoli", "mitoses", "outcome")

## Build the model for each
train_fit <- glm(selected_fit$formula, data = biopsy_train, family = "binomial")
test_fit <- glm(selected_fit$formula, data = biopsy_test, family = "binomial")

## Send to pROC::roc() function. Add arg quiet=T for shhhh
train_roc <- roc(biopsy_train$outcome, train_fit$linear.predictors, quiet = T)
test_roc  <- roc(biopsy_test$outcome, test_fit$linear.predictors, quiet = T)
```

```
train_roc$auc
## Area under the curve: 0.9969
test_roc$auc
## Area under the curve: 0.9916

train_data <- tibble(FPR = 1 - train_roc$specificities,
                     TPR = train_roc$sensitivities,
                     model = "Training split")
test_data <- tibble(FPR = 1 - train_roc$specificities,
                     TPR = train_roc$sensitivities,
                     model = "Testing split")

# Fiddled with size, color since lines are totally overlapping
bind_rows(train_data, test_data) %>%
  ggplot(aes(x = FPR, y = TPR, color = model)) +
    geom_line(aes(size = model)) +
    scale_color_manual(values = c("black", "yellow")) +
    scale_size_manual(values = c(2,1))
```



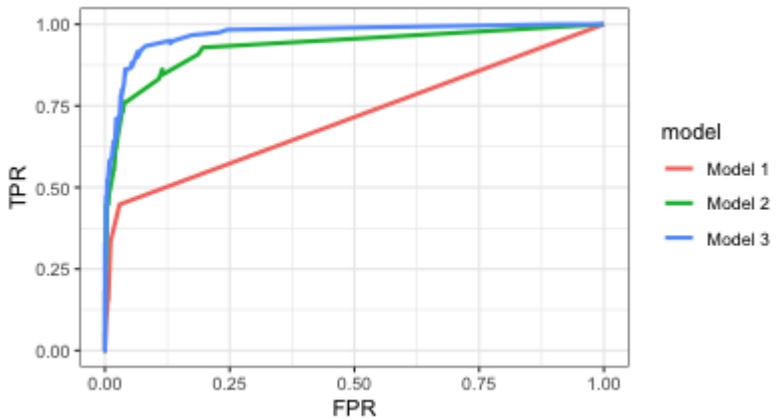
# Ok, let's get some "real life" going

```
model1 <- glm(outcome ~ mitoses, data = biopsy_fct, family = "binomial")
model1_roc <- roc(biopsy_fct$outcome, model1$linear.predictors, quiet = T)

model2 <- glm(outcome ~ mitoses + marg_adhesion, data = biopsy_fct, family =
"binomial")
model2_roc <- roc(biopsy_fct$outcome, model2$linear.predictors, quiet = T)

model3 <- glm(outcome ~ mitoses + marg_adhesion + epithelial_cell_size, data =
biopsy_fct, family = "binomial")
model3_roc <- roc(biopsy_fct$outcome, model3$linear.predictors, quiet = T)


model1_roc$auc
## Area under the curve: 0.7116
model2_roc$auc
## Area under the curve: 0.9308
model3_roc$auc
## Area under the curve: 0.9689
```

- Compared to Model 1...
  - **Model 2** shows that including **`marg_adhesion`** as predictor might add a LOT of benefit!
  - **Model 3** shows that including **`epithelial_cell_size`** as predictor might add even more benefit

# Compare ROC curves all three models

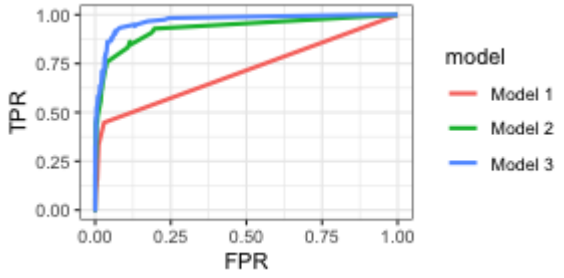
```
model1_data <- tibble(FPR = 1 - model1_roc$specificities,
                      TPR = model1_roc$sensitivities,
                      model = "Model 1")
model2_data <- tibble(FPR = 1 - model2_roc$specificities,
                      TPR = model2_roc$sensitivities,
                      model = "Model 2")
model3_data <- tibble(FPR = 1 - model3_roc$specificities,
                      TPR = model3_roc$sensitivities,
                      model = "Model 3")
bind_rows(model1_data, model2_data) %>%
  bind_rows(model3_data) %>%
  ggplot(aes(x = FPR, y = TPR, color = model)) +
    geom_line(size=1)
```



# Want to up your game?!!?

- Use functions **anytime you are writing the same code >=2x**
- Prevents bugs, cleaner to read, cleaner to reproduce

```
prep_roc <- function(roc_output, model_name){
  tibble(FPR = 1 - roc_output$specificities,
         TPR = roc_output$sensitivities,
         model = model_name)
}

model1_data <- prep_roc(model1_roc, "Model 1")
model2_data <- prep_roc(model2_roc, "Model 2")
model3_data <- prep_roc(model3_roc, "Model 3")

bind_rows(model1_data, model2_data) %>%
  bind_rows(model3_data) %>%
  ggplot(aes(x = FPR, y = TPR, color = model)) +
    geom_line(size=1)
```



# A note about tidy data

```
biopsy %>% head()
## # A tibble: 6 x 10
##   clump_thickness uniform_cell_si… uniform_cell_sh… marg_adhesion
##             <dbl>            <dbl>            <dbl>         <dbl>
## 1               5                1                1             1
## 2               5                4                4             5
## 3               3                1                1             1
## 4               6                8                8             1
## 5               4                1                1             3
## 6               8               10               10             8
## # … with 6 more variables: epithelial_cell_size <dbl>, bare_nuclei <dbl>,
## #   bland_chromatin <dbl>, normal_nucleoli <dbl>, mitoses <dbl>, outcome <chr>
```

```
biopsy %>%
  pivot_longer(clump_thickness:mitoses,
               names_to = "measurement", values_to = "value") %>%
  head()
## # A tibble: 6 x 3
##   outcome measurement          value
##   <chr>   <chr>                <dbl>
## 1 benign  clump_thickness          5
## 2 benign  uniform_cell_size        1
## 3 benign  uniform_cell_shape       1
## 4 benign  marg_adhesion            1
## 5 benign  epithelial_cell_size     2
## 6 benign  bare_nuclei              1
```