Introduction to logistic regression

Stephanie J. Spielman

Data Science for Biologists, Fall 2020

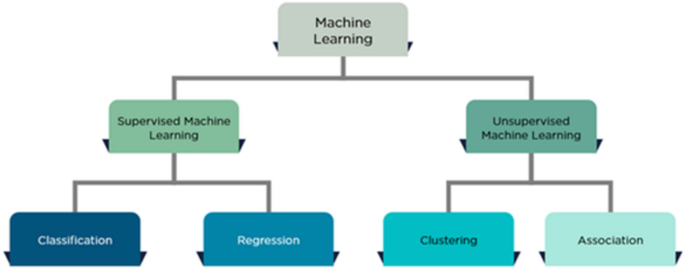
Linear regression vs. logistic regression

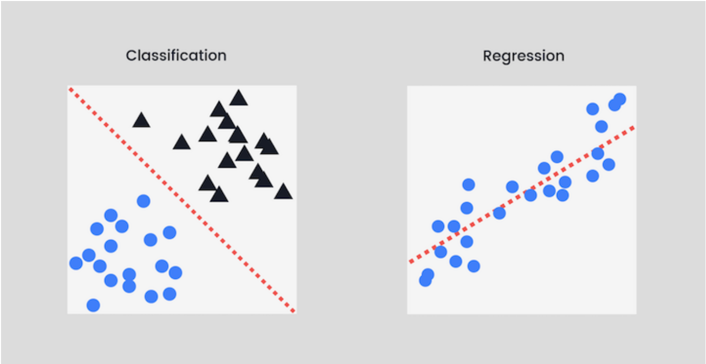
- Linear regression: How much do these (linearly-related) predictors explain variation in my *numeric* response variable?

- Logistic regression: How well do these predictors explain variation in my *categorical* ***binary*** response variable?
    - E.g. predicting Species in the iris dataset would be a categorical predictor, but NOT binary
    - Type of classifier

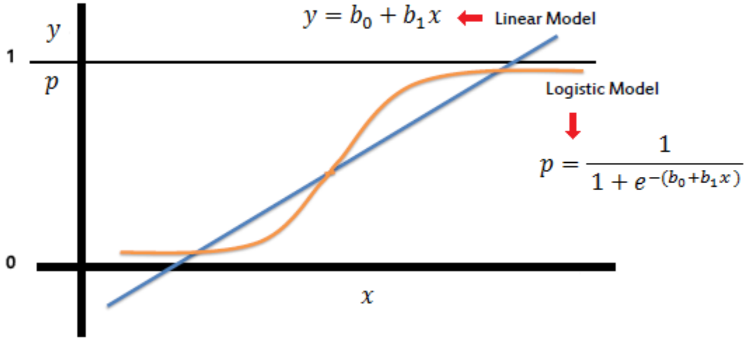
Where are we in the "machine learning" universe?

- Machine learning = the computer learns through experience
    - More data = more experience! *Training models on data IS machine learning*
    - Ignore the AI hype.



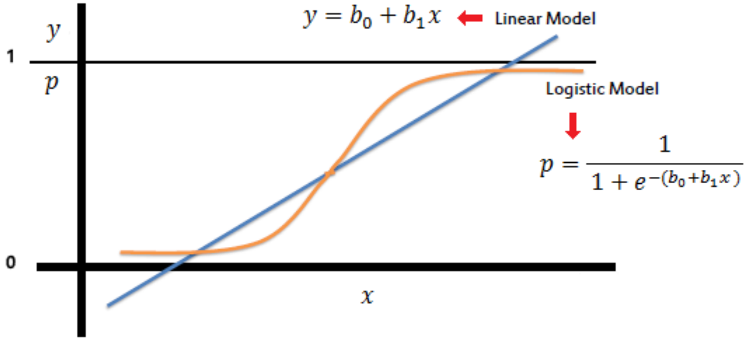


Logistic regression



- Linear regression: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 ... + \beta_N X_N + \epsilon$

Logistic regression



- Linear regression: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
- Logistic regression *transforms the predictors*
    - $t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
    - $Y = \frac{1}{1+e^{-t}}$ (or, $p = \ldots$ in image)

```
# too large to fit on slide..
data_url <- paste0("https://raw.githubusercontent.com/sjspielman/",
                   "datascience_for_biologists/master/docs/",
                   "fall2020/slides/biopsy.csv")

biopsy <- read_csv(data_url)

dplyr::glimpse(biopsy)
## Rows: 683
## Columns: 10
## $ clump_thickness      <dbl> …
## $ uniform_cell_size    <dbl> …
## $ uniform_cell_shape   <dbl> …
## $ marg_adhesion        <dbl> …
## $ epithelial_cell_size <dbl> …
## $ bare_nuclei          <dbl> …
## $ bland_chromatin      <dbl> …
## $ normal_nucleoli      <dbl> …
## $ mitoses              <dbl> …
## $ outcome              <chr> …
```

Building the logistic regression: Prepare the data

```
## Ensure the column is a factor, OR it has 0/1 values
## Help yourself by coding success = 1, failure = 0. This way you don't need
alphabetical order
biopsy %>%
  mutate(outcome_01 = case_when(outcome == "malignant" ~ 1, # "success"
                                outcome == "benign" ~ 0)) %>%
  select(-outcome) %>%
  select(outcome_01, everything()) -> biopsy_outcome01

head(biopsy_outcome01)
## # A tibble: 6 x 10
##   outcome_01 clump_thickness
##        <dbl>           <dbl>
## 1          0               5
## 2          0               5
## 3          0               3
## 4          0               6
## 5          0               4
## 6          1               8
## # … with 8 more variables:
## #   uniform_cell_size <dbl>,
## #   uniform_cell_shape <dbl>,
## #   marg_adhesion <dbl>,
## #   epithelial_cell_size <dbl>,
## #   bare_nuclei <dbl>,
## #   bland_chromatin <dbl>,
## #   normal_nucleoli <dbl>,
## #   mitoses <dbl>
```

Building the logistic regression: Build the model

`glm(response ~ predictors, data = data, family = "binomial")`

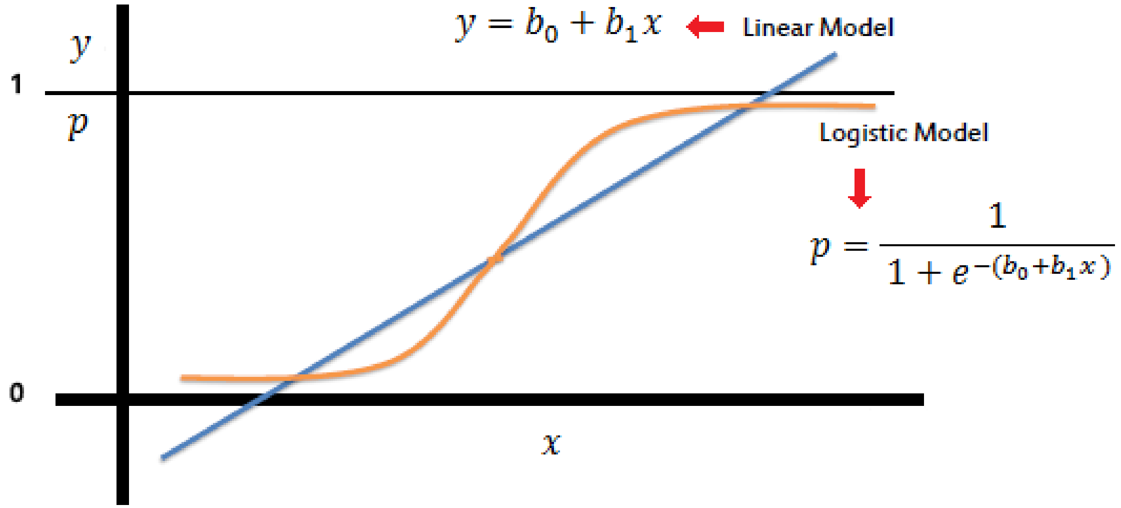
```
baseline_logit_fit <- glm(outcome_01 ~ ., data = biopsy_outcome01, family =
"binomial")

fit <- step(baseline_logit_fit, trace = F) # Read "Introduction to Model
Selection"!!
```

Interpreting the logistic regression coefficients

```
broom::tidy(fit)
## # A tibble: 8 x 5
##   term  estimate std.error
##   <chr>    <dbl>     <dbl>
## 1 (Int…   -9.98      1.13 
## 2 clum…    0.534     0.141
## 3 unif…    0.345     0.172
## 4 marg…    0.342     0.119
## 5 bare…    0.388     0.0936
## 6 blan…    0.462     0.168
## 7 norm…    0.226     0.111
## 8 mito…    0.531     0.324
## # … with 2 more variables:
## #   statistic <dbl>,
## #   p.value <dbl>
```

- For every unit increase in the predictor, the **log odds of success** of the response increases by the coefficient
  - $Pr(success)$ = probability of *malignant* biopsy for a given set of observations (predictors)
  - $Pr(failure)$ = probability of *benign* biopsy for a given set of observations
  - **Log odds** = $ln\left(\frac{Pr(success)}{Pr(failure)}\right)$

Visualizing the logistic regression



```
## USING head() to make it fit on slides!!

## What would have been your Y-values if this were regression
## YOUR X-AXIS !!
head(fit$linear.predictors)
##         1         2         3 
## -4.093622  2.032920 -4.773329 
##         4         5         6 
##  1.378604 -3.942642 10.636051

## The logit transformed - PROBABILITIES OF SUCCESS
## YOUR Y-AXIS !!
head(fit$fitted.values)
##           1           2 
## 0.016405105 0.884210413 
##           3           4 
## 0.008381356 0.798766714 
##           5           6 
## 0.019027825 0.999975967
## USING head() to make it fit on slides!!

## What would have been your Y-values if this were regression
## YOUR X-AXIS !!
head(fit$linear.predictors)
##         1         2         3
## -4.093622  2.032920 -4.773329
##         4         5         6
##  1.378604 -3.942642 10.636051

## The logit transformed - PROBABILITIES OF SUCCESS
## YOUR Y-AXIS !!
head(fit$fitted.values)
##           1           2
## 0.016405105 0.884210413
##           3           4
## 0.008381356 0.798766714
##           5           6
## 0.019027825 0.999975967
```

- $t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \ldots + \beta_N X_N + \epsilon$
- $Y = \frac{1}{1+e^{-t}}$

```
## USING head() to make it fit on slides!!

## What would have been your Y-values if this were regression
## YOUR X-AXIS !!
head(fit$linear.predictors)
##         1         2         3
## -4.093622  2.032920 -4.773329
##         4         5         6
##  1.378604 -3.942642 10.636051

## The logit transformed - PROBABILITIES OF SUCCESS
## YOUR Y-AXIS !!
head(fit$fitted.values)
##           1           2
## 0.016405105 0.884210413
##           3           4
## 0.008381356 0.798766714
##           5           6
## 0.019027825 0.999975967
```

- $t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 ... + \beta_N X_N + \epsilon$
- $Y = \frac{1}{1+e^{-t}}$

```
1/(1 + exp(-1 * fit$linear.predictors)) %>% head()
##           1           2
## 0.016405105 0.884210413
##           3           4
## 0.008381356 0.798766714
##           5           6
## 0.019027825 0.999975967
```

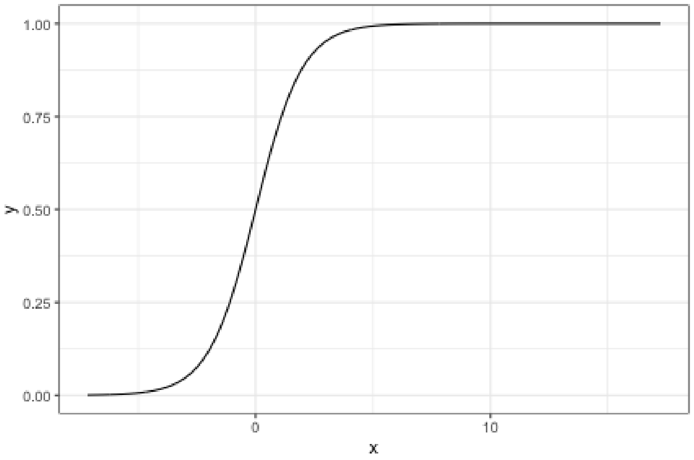
Visualizing the model: Prepare the data

```
tibble(x = fit$linear.predictors,
       y = fit$fitted.values,
       # Helps to use the ORIGINAL biopsy version so that outcome is
"malignant"/"benign"
       outcome = biopsy$outcome) -> fit_tibble

fit_tibble
## # A tibble: 683 x 3
##         x       y outcome
##     <dbl>   <dbl> <chr>
##   1 -4.09 0.0164  benign
##   2  2.03 0.884   benign
##   3 -4.77 0.00838 benign
##   4  1.38 0.799   benign
##   5 -3.94 0.0190  benign
##   6 10.6  1.00    malignant
##   7 -2.73 0.0609  benign
##   8 -5.35 0.00472 benign
##   9 -4.49 0.0110  benign
##  10 -5.09 0.00612 benign
## # … with 673 more rows
```

Visualizing the model

```
head(fit_tibble)
## # A tibble: 6 x 3
##        x       y outcome
##    <dbl>   <dbl> <chr>
## 1 -4.09 0.0164  benign
## 2  2.03 0.884   benign
## 3 -4.77 0.00838 benign
## 4  1.38 0.799   benign
## 5 -3.94 0.0190  benign
## 6 10.6  1.00    malignant
```

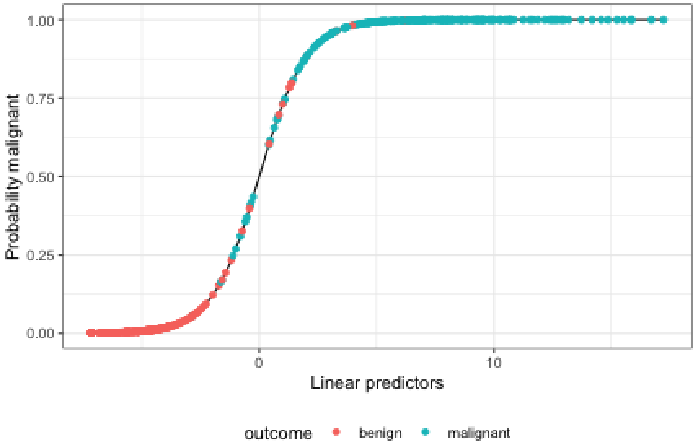
```
ggplot(fit_tibble, aes(x = x, y = y))
+
 geom_line() +
 theme(legend.position = "bottom")
```



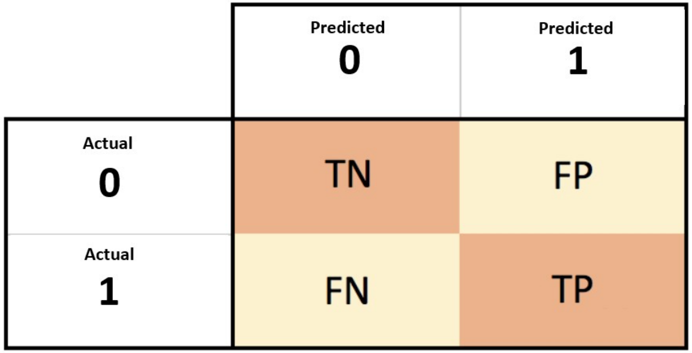
Visualizing the model FULLY!!!

```
head(fit_tibble)
## # A tibble: 6 x 3
##        x       y outcome
##    <dbl>   <dbl> <chr>
## 1 -4.09 0.0164  benign
## 2  2.03 0.884   benign
## 3 -4.77 0.00838 benign
## 4  1.38 0.799   benign
## 5 -3.94 0.0190  benign
## 6 10.6  1.00    malignant
```

```
ggplot(fit_tibble, aes(x = x, y = y))
+
 geom_line() +
geom_point(aes(color = outcome)) +
 theme(legend.position = "bottom") +
  labs(x = "Linear predictors",
       y = "Probability malignant")
```



Confusion matrix time



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

- **First ask:** is the result positive or negative? **Then ask:** should we have gotten that result though?
    - If yes, *TRUE*. If not, *FALSE*.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

A person using illegal performing enhancing drugs passes a test clearing them of drug use.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

A person using illegal performing enhancing drugs passes a test clearing them of drug use.

A study found a significant relationship between neck strain and jogging, when reality there is no relationship.

What is it?

A new arthritis drug does help pain clinical trials, even though it actually does reduce arthritis pain.

A person with HIV receives a positive test result for HIV.

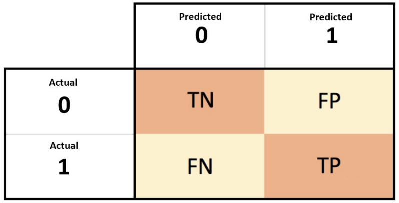
A person using illegal performing enhancing drugs passes a test clearing them of drug use.

A study found a significant relationship between neck strain and jogging, when reality there is no relationship.

A healthy individual gets a positive cancer biopsy result.

Classification metrics (an abbreviated set)

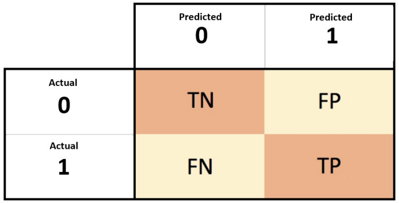
- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*



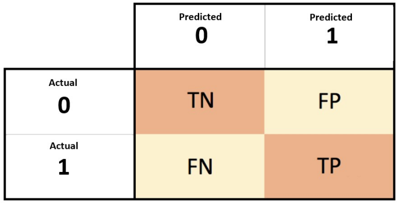
	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Classification metrics (an abbreviated set)

- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*

- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*

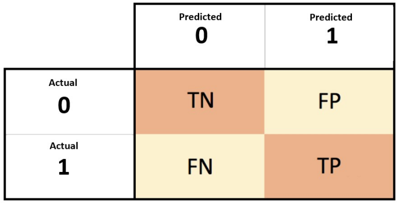
- False positive rate: $FPR = FP/N = \frac{FP}{FP+TN}$
  - AKA *1 - specificity*



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Classification metrics (an abbreviated set)

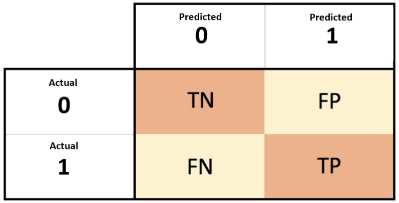
- True positive rate: $TPR = TP/P = \frac{TP}{TP+FN}$
  - AKA *sensitivity* AKA *recall*
- True negative rate: $TNR = TN/N = \frac{TN}{FP+TN}$
  - AKA *specificity*
- False positive rate: $FPR = FP/N = \frac{FP}{FP+TN}$
  - AKA *1 - specificity*
- Precision: $PPV = \frac{TP}{TP+FP}$
  - AKA *positive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



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Classification metrics (an abbreviated set)

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



	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Caculating performance measures

- Requires a *threshold* to call malignant/benign outcomes.
- For an example, let's say >=0.75 is malignant (success). <0.75 is benign (failure)
- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$

```
# Reminder:
tibble(x = fit$linear.predictors,
       y = fit$fitted.values,
       outcome = biopsy$outcome) -> fit_tibble

fit_tibble
## # A tibble: 683 x 3
##         x       y outcome
##     <dbl>   <dbl> <chr>
##   1 -4.09 0.0164  benign
##   2  2.03 0.884   benign
##   3 -4.77 0.00838 benign
##   4  1.38 0.799   benign
##   5 -3.94 0.0190  benign
##   6 10.6  1.00    malignant
##   7 -2.73 0.0609  benign
##   8 -5.35 0.00472 benign
##   9 -4.49 0.0110  benign
##  10 -5.09 0.00612 benign
## # … with 673 more rows
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

```
threshold <- 0.75
fit_tibble %>%
  rename(truth = outcome) %>%
  mutate(pred = if_else(y >= threshold, "P", "N"))
## # A tibble: 683 x 4
##         x       y truth   pred 
##     <dbl>   <dbl> <chr>   <chr>
##   1 -4.09 0.0164  benign  N    
##   2  2.03 0.884   benign  P    
##   3 -4.77 0.00838 benign  N    
##   4  1.38 0.799   benign  P    
##   5 -3.94 0.0190  benign  N    
##   6 10.6  1.00    malign… P    
##   7 -2.73 0.0609  benign  N    
##   8 -5.35 0.00472 benign  N    
##   9 -4.49 0.0110  benign  N    
## 10 -5.09 0.00612 benign  N    
## # … with 673 more rows
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

```
threshold <- 0.75
fit_tibble %>%
  rename(truth = outcome) %>%
  mutate(pred = if_else(y >= threshold, "P", "N")) %>%
  mutate(classif = case_when(truth == "malignant" & pred == "P" ~ "TP",
                             truth == "malignant" & pred == "N" ~ "FN",
                             truth == "benign"    & pred == "N" ~ "TN",
                             truth == "benign"    & pred == "P" ~ "FP")) ->
model_classif

model_classif
## # A tibble: 683 x 5
##          x       y truth    pred
##      <dbl>   <dbl> <chr>    <chr>
##  1 -4.09 0.0164  benign   N
##  2  2.03 0.884   benign   P
##  3 -4.77 0.00838 benign   N
##  4  1.38 0.799   benign   P
##  5 -3.94 0.0190  benign   N
##  6 10.6  1.00    malign… P
##  7 -2.73 0.0609  benign   N
##  8 -5.35 0.00472 benign   N
##  9 -4.49 0.0110  benign   N
## 10 -5.09 0.00612 benign   N
## # … with 673 more rows, and 1
## #   more variable:
## #   classif <chr>
```

```
model_classif %>%
  # how many in each classif category?
  count(classif)
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- Accuracy = (437 + 219) / (20 + 7 + 437 + 219) = **0.96**

```
model_classif %>%
  # how many in each classif category?
  count(classif)
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

- Accuracy = (437 + 219) / (20 + 7 + 437 + 219) = **0.96**

```
model_classif %>%
  count(classif) %>%
  pivot_wider(names_from = classif,
values_from = n)
## # A tibble: 1 x 4
##      FN    FP    TN    TP
##   <int> <int> <int> <int>
## 1    20     7   437   219
```

```
model_classif %>%
  count(classif) %>%
  pivot_wider(names_from = classif,
values_from = n) %>%
  mutate(accuracy = (TP + TN)/(TP +
TN + FP + FN))
## # A tibble: 1 x 5
##      FN    FP    TN    TP
##   <int> <int> <int> <int>
## 1    20     7   437   219
## # … with 1 more variable:
## #   accuracy <dbl>
```

How good is the model?

- In linear regression, we often uses R^2 values to compare different viable models. Higher R^2 often (but not always!) means, "more predictive model"

- In logistic regression, performance **depends** on your chosen threshold! So, how do we choose a threshold?
  - Usually, find the threshold that makes the false positive rate <5%>
- We also use **AUC** (area under the curve… what curve?)