Introduction to logistic regression

Stephanie J. Spielman

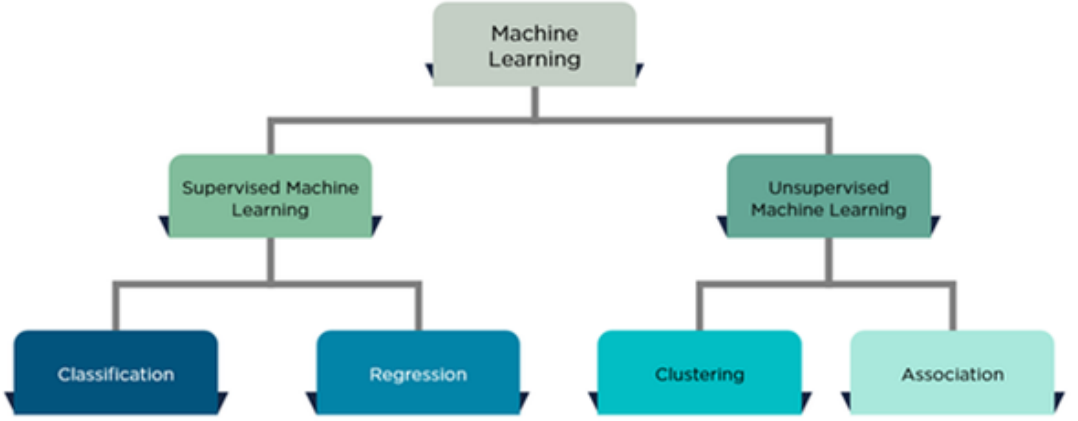
Data Science for Biologists, Spring 2020

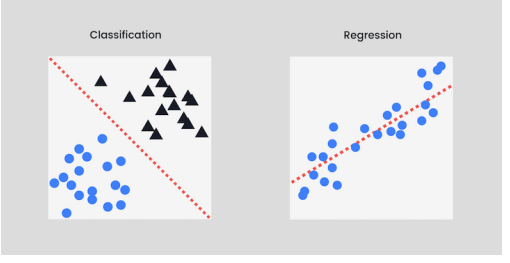
Linear regression vs. logistic regression

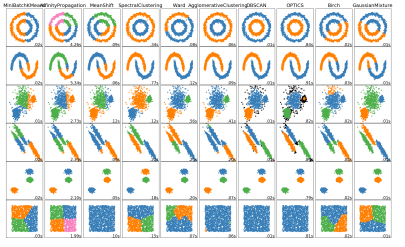
- Linear regression: How much do these (linearly-related) predictors explain variation in my *numeric* response variable?
- Logistic regression: How well do these predictors explain variation in my *categorical* ***binary*** response variable?
  - E.g. predicting Species in the iris dataset would be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

Where are we in the "machine learning" universe?

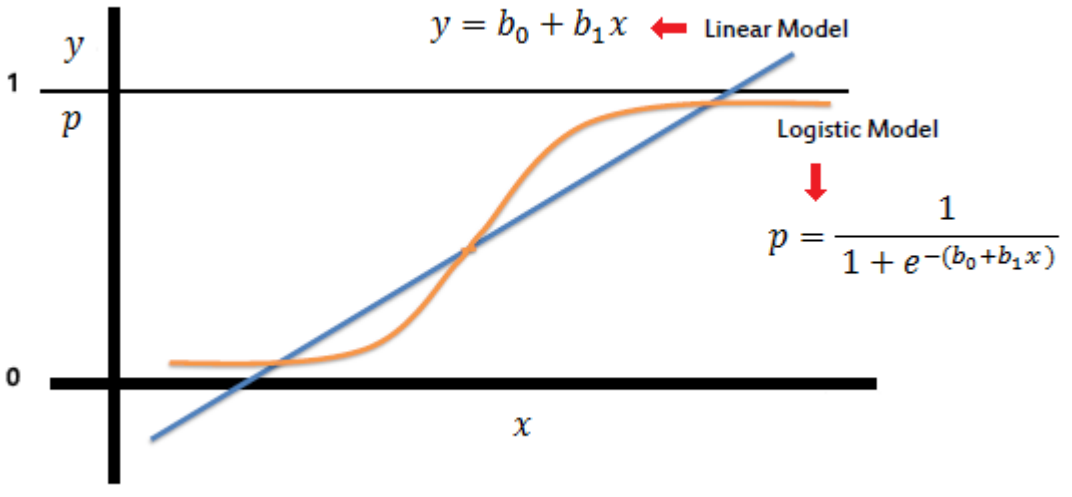
- Machine learning = the computer learns through experience
  - More data = more experience! *Training models on data IS machine learning*
  - Ignore the AI hype.







Logistic regression



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

```
# hacking to fit URL on the slide...
biopsy <- read_csv(
          paste0("https://raw.githubusercontent.com/sjspielman/",
                 "datascience_for_biologists/master/slides/biopsy.csv"))
## Parsed with column specification:
## cols(
##   clump_thickness = col_double(),
##   uniform_cell_size = col_double(),
##   uniform_cell_shape = col_double(),
##   marg_adhesion = col_double(),
##   epithelial_cell_size = col_double(),
##   bare_nuclei = col_double(),
##   bland_chromatin = col_double(),
##   normal_nucleoli = col_double(),
##   mitoses = col_double(),
##   outcome = col_character()
## )

dplyr::glimpse(biopsy)
## Rows: 683
## Columns: 10
## $ clump_thickness      <dbl> 5, 5, 3, 6, 4, 8, 1, 2, 2, 4, 1, 2, 5, 1, 8, 7, …
## $ uniform_cell_size    <dbl> 1, 4, 1, 8, 1, 10, 1, 1, 1, 2, 1, 1, 3, 1, 7, 4,…
## $ uniform_cell_shape   <dbl> 1, 4, 1, 8, 1, 10, 1, 2, 1, 1, 1, 1, 3, 1, 5, 6,…
## $ marg_adhesion        <dbl> 1, 5, 1, 1, 3, 8, 1, 1, 1, 1, 1, 1, 3, 1, 10, 4,…
## $ epithelial_cell_size <dbl> 2, 7, 2, 3, 2, 7, 2, 2, 2, 2, 1, 2, 2, 2, 7, 6, …
## $ bare_nuclei          <dbl> 1, 10, 2, 4, 1, 10, 10, 1, 1, 1, 1, 1, 3, 3, 9, …
## $ bland_chromatin      <dbl> 3, 3, 3, 3, 3, 9, 3, 3, 1, 2, 3, 2, 4, 3, 5, 4, …
## $ normal_nucleoli      <dbl> 1, 2, 1, 7, 1, 7, 1, 1, 1, 1, 1, 1, 4, 1, 5, 3, …
## $ mitoses              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 4, 1, …
## $ outcome              <chr> "benign", "benign", "benign", "benign", "benign"…
```

Building the logistic regression

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

```
## Ensure the column is a factor, OR it's 0/1 values
## Help yourself by coding success = 1, failure = 0. This way you don't need
alphabetical order
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)
```

Interpreting the logistic regression coefficients

```
broom::tidy(selected_fit)
## # A tibble: 8 x 5
##   term               estimate std.error statistic  p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)          -9.98     1.13       -8.86 7.66e-19
## 2 clump_thickness       0.534    0.141       3.79 1.49e- 4
## 3 uniform_cell_shape    0.345    0.172       2.01 4.43e- 2
## 4 marg_adhesion         0.342    0.119       2.87 4.07e- 3
## 5 bare_nuclei           0.388    0.0936      4.15 3.32e- 5
## 6 bland_chromatin       0.462    0.168       2.75 6.02e- 3
## 7 normal_nucleoli       0.226    0.111       2.04 4.16e- 2
## 8 mitoses               0.531    0.324       1.64 1.02e- 1
```

- For every unit increase in the predictor, the **log odd 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** = $ln\left(\frac{Pr(success)}{Pr(failure)}\right)$

Using output from 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
selected_fit$linear.predictors %>% head()
##         1         2         3         4         5         6
## -4.093622  2.032920 -4.773329  1.378604 -3.942642 10.636051

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

Using output from 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
selected_fit$linear.predictors %>% head()
##         1         2         3         4         5         6
## -4.093622  2.032920 -4.773329  1.378604 -3.942642 10.636051

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

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

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

An option with `broom`, if you dare!

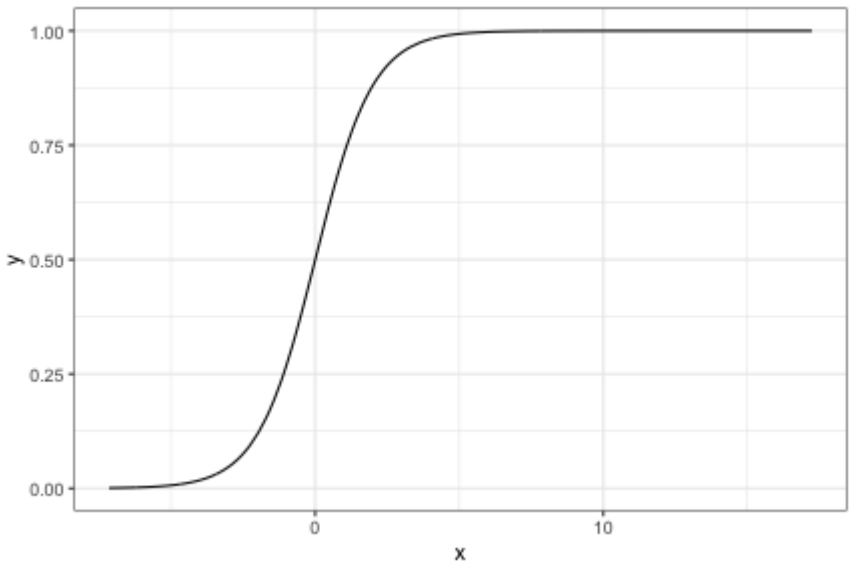
- The `.fitted` column is the x-axis in logit, need to transform directly for y

```
broom::augment(selected_fit) %>%
  select(outcome, .fitted) %>%
  rename(linear_predictors = .fitted) %>%
  mutate(probabilities     = 1/(1 + exp(-1 * linear_predictors)) )
## # A tibble: 683 x 3
##    outcome linear_predictors probabilities
##      <dbl>             <dbl>         <dbl>
##  1       0             -4.09       0.0164
##  2       0              2.03       0.884
##  3       0             -4.77       0.00838
##  4       0              1.38       0.799
##  5       0             -3.94       0.0190
##  6       1             10.6        1.00
##  7       0             -2.73       0.0609
##  8       0             -5.35       0.00472
##  9       0             -4.49       0.0110
## 10       0             -5.09       0.00612
## # … with 673 more rows
```

Visualizing the model

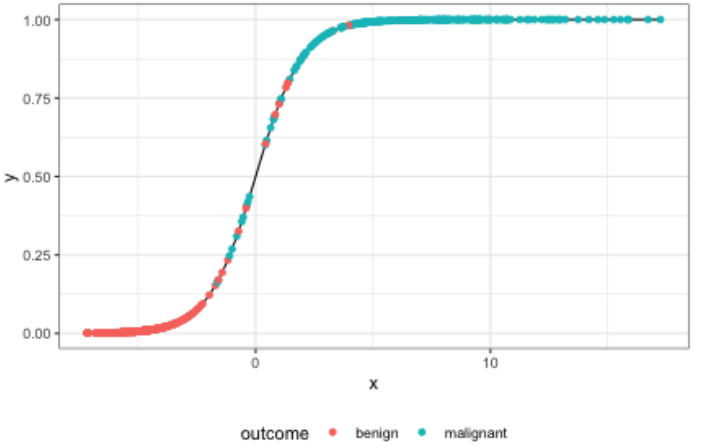
```
tibble(x = selected_fit$linear.predictors,
       y = selected_fit$fitted.values,
       outcome = biopsy$outcome) %>%
  ggplot(aes(x = x, y = y)) +
    geom_line() +
    theme(legend.position = "bottom")-> plot_of_model

plot_of_model
```



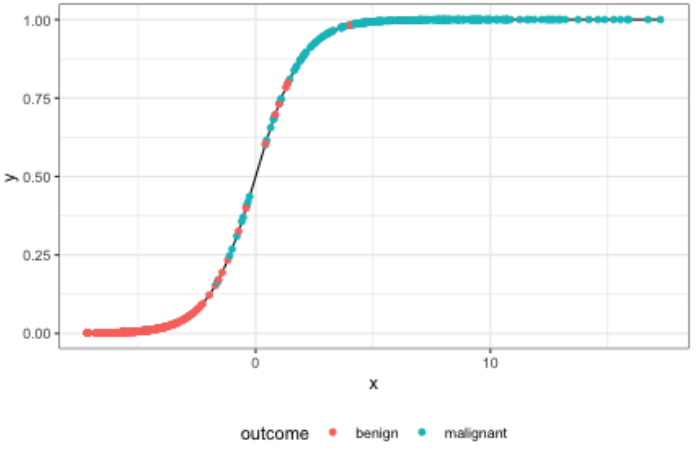
Visualizing the model

```
plot_of_model +
  geom_point(aes(color = outcome))
```

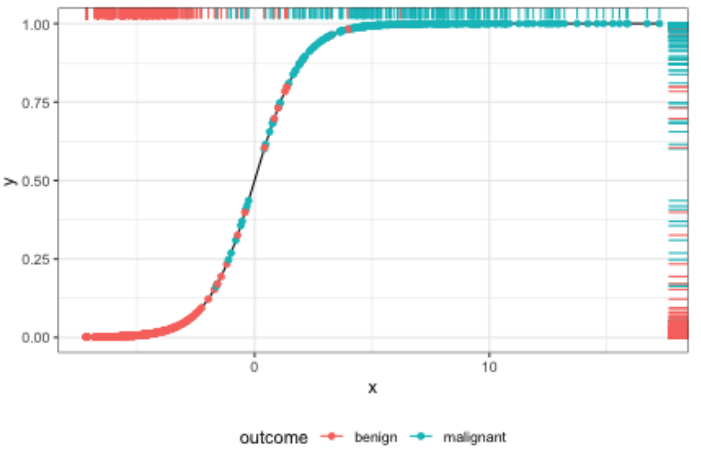


Visualizing the model

```
plot_of_model +
  geom_point(aes(color = outcome))
```




```
plot_of_model +
  geom_point(aes(color = outcome)) +
  geom_rug(sides = "tr", aes(color =
outcome))
```



Visualizing the model

```
tibble(x = selected_fit$linear.predictors,
       outcome = biopsy$outcome) %>%
  ggplot(aes(x = x, fill = outcome)) +
    geom_density(alpha = 0.6)
```

Confusion matrix time

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

Confusion matrix time

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

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

Confusion matrix time

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

- 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.

Confusion matrix time

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

- 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.

Confusion matrix time

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

- 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.

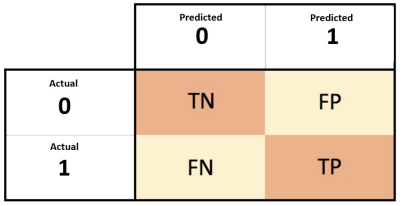
Confusion matrix time

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

- 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.

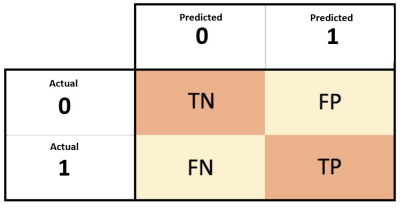
Classification metrics (an abbreviated set)

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



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*



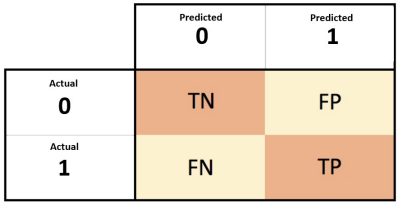
Classification metrics (an abbreviated set)

- 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*



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: $\frac{TP+TN}{TP+TN+FP+FN}$

```
tibble(x = selected_fit$linear.predictors,
       y = selected_fit$fitted.values,
       outcome = biopsy$outcome) -> model_tibble

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

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

```
threshold <- 0.75
model_tibble %>%
  rename(truth = outcome) %>%
  mutate(pred = if_else(y >= threshold, "pos", "neg"))
```

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

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

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

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

```
model_classif %>%
  count(classif) #<< short for `group_by(classif) %>% tally()`
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

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

```
model_classif %>%
  count(classif) #<< short for `group_by(classif) %>% tally()`
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

- 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) #<< short for `group_by(classif) %>% tally()`
## # A tibble: 4 x 2
##   classif     n
##   <chr>   <int>
## 1 FN         20
## 2 FP          7
## 3 TN        437
## 4 TP        219
```

- 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 accuracy
##   <int> <int> <int> <int>    <dbl>
## 1    20     7   437   219    0.960
```

What did we learn today?

- What is logistic regression?
- How to perform and visualize logistic regression
  - **Use `glm()` NOT `lm()`**
  - **Do not forget to add the argument `family="binomial"`**
- How to classify basic performance at a given threshold

What did we learn today?

- What is logistic regression?
- How to perform and visualize logistic regression
  - **Use `glm()` NOT `lm()`**
  - **Do not forget to add the argument `family="binomial"`**
- How to classify basic performance at a given threshold

- **Next up**
  - What about any threshold???
  - ROC curve and AUC as performance evaluators
  - Testing/training splits (code too gross for other cross validation during Remote Times)