# CWRU DSCI351-351M-451: Week15a-p Logistic Regression

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

01 December, 2022

## Contents

#### 15.1.2.1 Class Readings, Assignments, Syllabus Topics

##### 15.1.2.1.1 Course Evaluations Are Open Now

- lets get to 90% response rate
- We want statistically significant results!
  - I look for suggestions on how to improve the course
- https://webapps.case.edu/courseevals/

##### 15.1.2.1.2 Reading, Lab Exercises, SemProjects

- Readings:
  - For today: ISLR 3.1,3.2
  - For next class: French & Bruckman 2020
- Laboratory Exercises:
  - LE7 : Due Thursday Dec. 8nd
  - LE7 :
- Office Hours: (Class Canvas Calendar for Zoom Link)
  - Wednesday @ 4:00 PM to 5:00 PM, Will Oltjen
  - Saturday @ 3:00 PM to 4:00 PM, Kristen Hernandez
  - **Office Hours are on Zoom, and recorded**
- Semester Projects
  - DSCI 451 Students Biweekly Update 6 Due Friday November 18th
  - DSCI 451 Students
    - Next
  - All DSCI 351/351M/451 Students:
    - **Peer Grading of Report Out #3 is Given out today**
  - Exams
    - Final: Monday December 19, 2022, 12:00PM - 3:00PM, Nord 356 or remote

#### 15.1.2.2 Syllabus

### 15.1.3 Logistic Regression

```
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(ggplot2)
library(broom)
```

| Day:Date | Foundation | Practicum | Reading | Due |
|---|---|---|---|---|
| w01a:Tu:8/30/22 | ODS Tool Chain | R, Rstudio, Git | | |
| w01b:Th:9/1/22 | Setup ODS Tool Chain | Bash, Git, Slack, Agile | PRP4-33 | LE1 |
| w02a:Tu:9/6/22 | Bash-Git-Knuth-Lit.Prog. | RIntroR | PRP35-64 | |
| w02b:Th:9/8/22 | What is Data Science | OIS:Intro2R | OIS1,2 | |
| w02Pr:Fr:9/9/22 | | | PRP65-93 | **451 Update1** |
| w03a:Tu:9/13/22 | Data Intro | Data Analytic Style | PRP94-116 | LE2 **LE1 Due** |
| w03b:Th:9/15/22 | Rand. Var. Normal Dist. | Git, Rmds, Loops | OIS4 | |
| w04a:Tu:9/20/22 | Tidy Check Explore | Tidy GapMinder | EDA1-31 | |
| w04b:Th:9/22/22 | Inference, DSCI Process | Other Distrib. 7 ways | R4DS1-3 | LE3 **LE2 Due** |
| w04Pr:Fr:9/23/22 | | | EDA32-58 | **451 Update2** |
| w05a:Tu:9/27/22 | OIS4 Rand. Var. | EDA of PET Degr. | OIS5 | |
| w05b:Th:9/29/22<br>w05Pr:Fr:9/30/22 | OIS5 Found. of Infer. | Multivar Corr. Plot | R4DS4-6 | <br>**451 RepOut1** |
| w06a:Tu:10/4/22 | Pred., Algorithm, Model | | R4DS7-8 | |
| w06b:Th:10/6/22 | Summ. Stats & Vis. | Anscombe's Quartets | R4DS9-16 | LE4 **LE3 Due** |
| w06Pr:Fr:10/7/22 | | | | **451 Update3** |
| w07a:Tu:10/11/22 | Midterm Rev. Tidy Data | Correl Plots Summ Stats | OIS6.1-2 | **PeerRv1 Due** |
| w07b:Th:10/13/22 | HypoTest, Infer. Recap | Penguin EDA, Sampling | | |
| w08a:Tu:10/18/22<br>w08b:Th:10/20/22 | **MIDTERM**<br>Programming & Coding | **EXAM**<br>Code Packaging | | <br>**LE4 Due** |
| w08Pr:Fr:10/21/22 | | | | **451 Update4** |
| Tu:10/24,25 | **CWRU** | **FALL BREAK** | R4DS17-21 | |
| w09b:Th:10/27/22 | Cat. Inf. 1 & 2 propor. | Indep. Test,2-way tables | OIS6.3-4 | LE5 |
| w09Pr:Fr:10/28/22 | | | | **451 RepOut2** |
| w10a:Tu:11/1/22 | Goodness of Fit, $\chi^2$ test | t-tests 1&2 means | OIS7.1-4 | |
| w10b:Th:11/3/22<br>w10Pr:Fr:11/4/22 | Num. Infer, Cont. Tables | Stat. Power | | <br>**451 Update5** |
| w11a:Tu:11/8/22<br>w11b:Th:11/10/22 | Sample & Effect Size<br>Regr Part 1, Test & Train | Stat. Power GGmap<br>Curse of Dimen. | OIS8<br>ISLR1,2.1,2 | **PeerRv2 Due**<br>LE6 **LE5 Due** |
| w12a:Tu:11/15/22 | Regr. Outliers | Regr Part 2, GIS | OIS9 | |
| w12b:Th:11/17/22 | Mult.Regr., Var. Select | Regr. Diagnostics | | |
| w12Pr:Fr:11/18/22 | | | | **451 Update6** |
| w13a:Tu:11/22/22 | Log. Regr. | Mult. Regression | ISLR3.1 | LE7 **LE6 due** |
| w13b:Th:11/24/22 | Statistical learning | Logistic Regr. | ISLR3.2 | |
| w13Pr:Fr:11/25/22 | | | | **451 RepOut3** |
| w14a:Tu:11/23/22 | | GIS Trends | ISLR4.1-3 | |
| Th,Fr:11/24,25 | **THANKSGIVIING** | **Vacation** | | |
| w15a:Tu:11/29/22 | Classificat., Sup. Lrning | Log. Regr. & ML | | **PeerRv3 Due** |
| w15b:Th:12/1/22 | Clustering, Unsup. Lrning | Caret, Broom 4 modeling | Fr.Br.2020 | |
| w15SPr:Fr:12/2/22 | | | | |
| w16a:Tu:12/6/22 | Big Data Analytics | Dist. Comp., Hadoop | Khalil.2020 | |
| w16b:Th:12/8/22 | Final Exam Review | | Mirletz,2015 | **LE7 due** |
| **Friday 12/12** | **SemProj** | **Final Report** | | **SemProj4 due** |
| **Monday 12/19** | **FINAL EXAM** | **12:00-3:00pm** | Nord 356 | or remote |

Figure 1: DSCI351-351M-451 Syllabus

```
library(forcats)
library(caret)
```

```
## Loading required package: lattice
```

##### 15.1.3.1 What is, Preparing data for, and how to evaluate, logistic regression

##### 15.1.3.2 Logistic regression theory

###### 15.1.3.2.1 What is a logistic regression?

- A logistic regression is a linear regression,
  - applied to categorical outcomes
  - by using the "logit", or log odds, transformation function.

###### 15.1.3.2.2 A linear regression

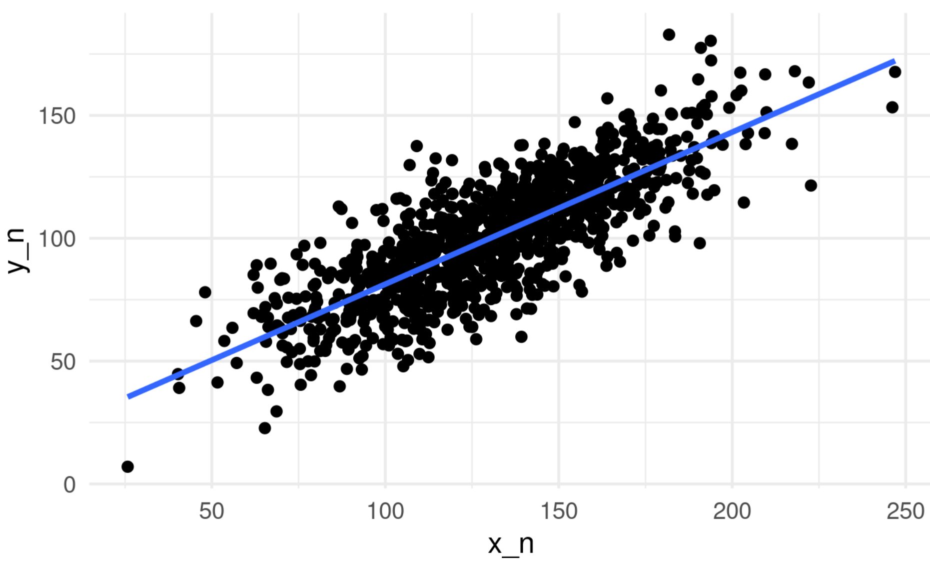
- A linear regression
  - uses a *line of best fit*
    - (the old $y = mx + c$)
    - what we would call $Y = \beta_0 + \beta_1 X + \epsilon$
  - over multiple variables to predict a continuous variable.

{Are you familiar with `qplot` command?}

```
set.seed(777)
y_n <- rnorm(1000, 100, 25)
x_n <- y_n + rnorm(1000, 30, 20)

?qplot
qplot(x_n, y_n) + geom_smooth(method = "lm", se = FALSE) + theme_minimal()
```

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

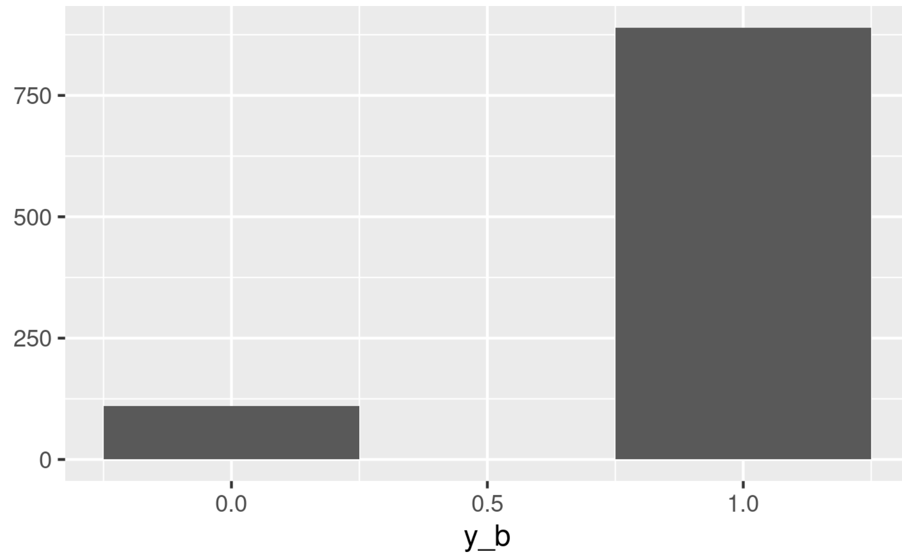



###### 15.1.3.2.3 Why do we need a transformation function?

- If you're trying to predict

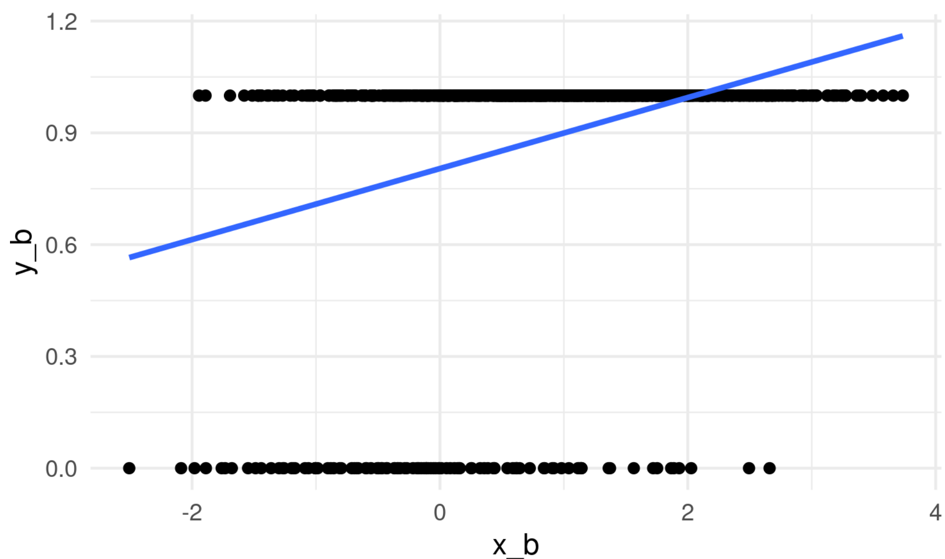
  - whether someone survives (1) or dies (0),
  - does it make sense to say they're
    * -0.2 alive, 0.5 alive, or 1.1 alive?

```
y_b <- rbinom(1000, size = 1, prob = .89)
qplot(y_b, binwidth = .5)
```




```
x_b <- y_b + rnorm(1000)
qplot(x_b, y_b) + geom_smooth(method = "lm", se = FALSE) + theme_minimal()
```

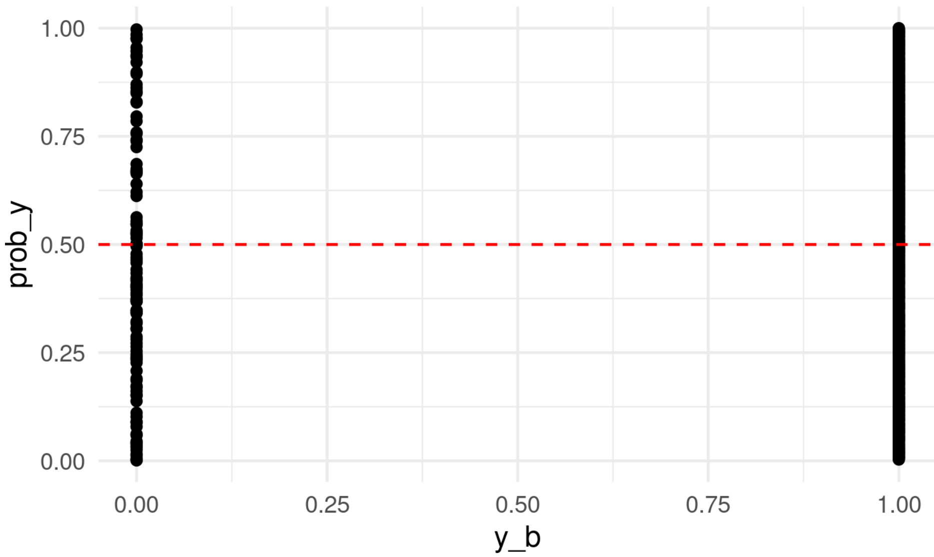
```
## `geom_smooth()` using formula = 'y ~ x'
```




##### 15.1.3.2.4 What can we measure that is a continuous variable?

- We can measure the *probability* of someone surviving.
  - This gives us data in the range $[0, 1]$
    * which is better,
  - but still not our ideal of $[-\infty, +\infty]$.

```
prob_y <- seq(0, 1, by = .001)[-1]
qplot(y_b, prob_y) + theme_minimal() + geom_hline(aes(yintercept = .5),
                                                  linetype = "dashed",
                                                  colour = "red")
```




#### 15.1.3.2.5 How can we transform it to be in the range we want?

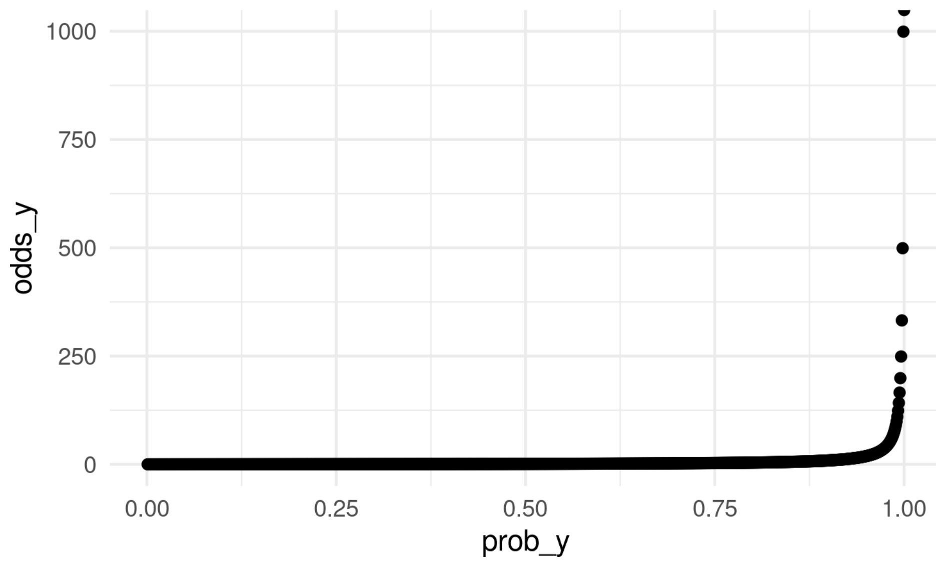
- The *odds* of something happening are
    - the probability of it happening versus
    - the probability of it not happening can help us.

$$\frac{p}{1-p}$$

As probability can never be less than 0 or greater than 1,

- we get a range between $[0, +\infty]$.

```
odds_y <- prob_y / (1 - prob_y)
qplot(prob_y, odds_y) + theme_minimal()
```

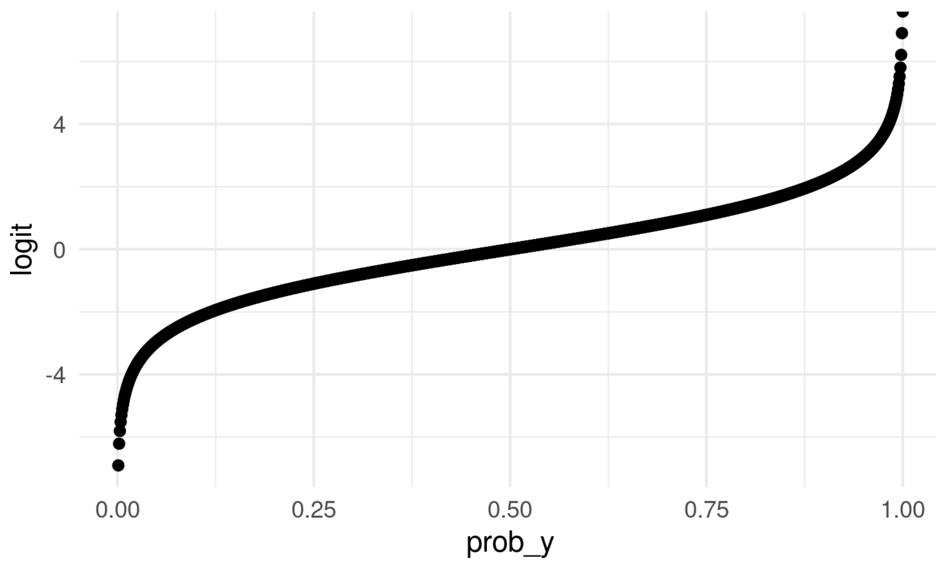



##### 15.1.3.2.6 How can allow negative values?

- The final step in this transformation
  - is to take the log of the odds,
  - which is commonly called the *logit*.

This gets us to $[-\infty, +\infty]$.

```
logit <- log(odds_y)
qplot(prob_y, logit) + theme_minimal()
```




```
#install.packages("optiRum")
library(optiRum)
logits     <- -4:4
odds       <- logit.odd(logits)
probs      <- odd.prob(odds)
pred_class <- logits >= 0
```

```
knitr::kable(data.frame(logits, odds, probs, pred_class))
```

##### 15.1.3.2.7 Interpreting the results

| logits | odds | probs | pred_class |
|---:|---:|---:|:---|
| -4 | 0.0183156 | 0.0179862 | FALSE |
| -3 | 0.0497871 | 0.0474259 | FALSE |
| -2 | 0.1353353 | 0.1192029 | FALSE |
| -1 | 0.3678794 | 0.2689414 | FALSE |
| 0 | 1.0000000 | 0.5000000 | TRUE |
| 1 | 2.7182818 | 0.7310586 | TRUE |
| 2 | 7.3890561 | 0.8807971 | TRUE |
| 3 | 20.0855369 | 0.9525741 | TRUE |
| 4 | 54.5981500 | 0.9820138 | TRUE |

#### 15.1.3.3 Logistic regressions in R

##### 15.1.3.3.1 `glm()` "generalized linear models"

- The `glm` function is used for performing logistic regressions.

It can be used for other linear models too.

```
?glm
glm(vs ~ mpg , data = mtcars, family = binomial(link = "logit"))
```

```
##
## Call:  glm(formula = vs ~ mpg, family = binomial(link = "logit"), data = mtcars)
##
## Coefficients:
## (Intercept)          mpg
##     -8.8331       0.4304
##
## Degrees of Freedom: 31 Total (i.e. Null);  30 Residual
## Null Deviance:       43.86
## Residual Deviance: 25.53     AIC: 29.53
```

##### 15.1.3.3.2 Formula

- R uses a formula system for specifying a model.
    - You put the outcome variable on the left
    - A tilde (`~`) is used for saying "predicted by"
    - Exclude an intercept term by adding `-1` to your formula
    - You can use a `.` to predict by all other variables e.g. `y ~ .`
    - Use a `+` to provide multiple independent variables e.g. `y ~ a + b`
    - You can use a `:` to use the interaction of two variables e.g. `y ~ a:b`
    - You can use a `*` to use two variables and their interaction e.g. `y ~ a*b`
        * (equivalent to `y ~ a + b + a:b`)
    - You can construct features on the fly
        * e.g. `y ~ log(x)` -or use `I()` when adding values
        * e.g. `y ~ I(a+b)`

For more info, check out `?formula`

#### 15.1.3.3.3 Useful parameters

- `na.action` can be set to amend the handling of missing values in the data
- `model`,`x`,`y` controls whether you get extra info about the model and data back.
  - Setting these to `FALSE` saves space

```
df <-
  data.frame(
    Function = c(
      "coefficients",
      "summary",
      "fitted",
      "predict",
      "plot",
      "residuals"
    ),
    Purpose = c(
      "Extract coefficients",
      "Output a basic summary",
      "Return the predicted values for the training data",
      "Predict some values for new data",
      "Produce some basic diagnostic plots",
      "Return the errors on predicted values for the training data"
    )
  )
knitr::kable(df)
```

#### 15.1.3.3.4 Functions working with `glm`

| Function | Purpose |
|---|---|
| coefficients | Extract coefficients |
| summary | Output a basic summary |
| fitted | Return the predicted values for the training data |
| predict | Predict some values for new data |
| plot | Produce some basic diagnostic plots |
| residuals | Return the errors on predicted values for the training data |

```
# kable is a simple way to make good looking tables in Rmd
?knitr::kable
```

#### 15.1.3.3.5 Inputs

- You can provide a `glm` with continuous and categorical variables.
  - Categorical variables get transformed into indicator (dummy) variables
  - Continuous variables should ideally be scaled

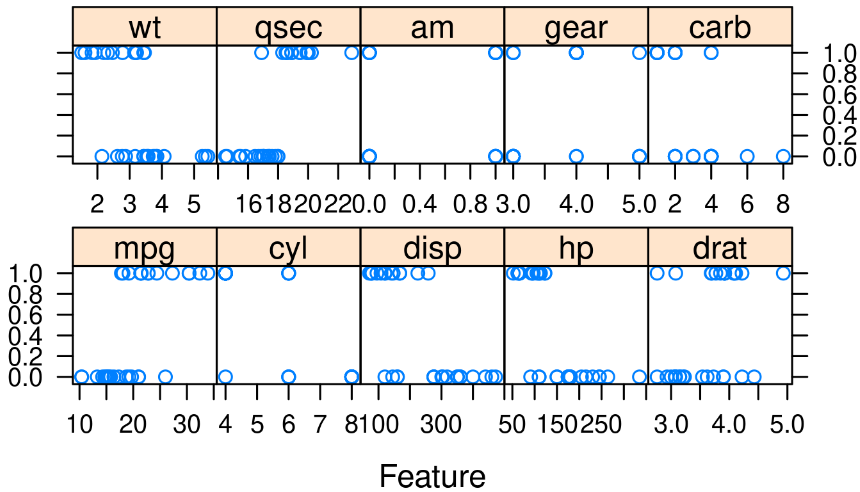
### 15.1.3.4 Preparing data

#### 15.1.3.4.1 Exploration

- Many ways to explore your data for outliers, patterns, issues etc.

```r
mtcarsVars <- mtcars[, colnames(mtcars)[colnames(mtcars) != "vs"]]
mtcarsOut <- mtcars[, "vs"]
```

```r
library(caret)
featurePlot(mtcarsVars, mtcarsOut)
```




#### 15.1.3.4.2 Sampling

- Commonly, we will take a training sample and a testing sample.

```r
set.seed(77887)
trainRows <- createDataPartition(mtcarsOut, p = .7 , list = FALSE)
training_x <- mtcarsVars[trainRows,]
training_y <- mtcarsOut[trainRows]
testing_x <- mtcarsVars[-trainRows,]
testing_y <- mtcarsOut[-trainRows]
```

#### 15.1.3.4.3 Why sample *before* processing?

- Sampling before scaling etc
  - prevents information about the test data leaking into our model.

By preventing such leaks

- we get a truer view of how well our model generalizes later.

#### 15.1.3.4.4 Scaling variables

- **minmax** Express numbers
  - as a percentage of the maximum
    - after subtracting the minimum.

This results in range $[0, 1]$

- for training data
  - but can result in a different range in test data
  - and, therefore, production!

$$\frac{x - min(x)}{max(x) - min(x)}$$

- **z-score** Express numbers
    - as the distance from the mean
        * in standard deviations.

This results in a range that's notionally $[-\infty, +\infty]$

- and results will be in the same range in test data.

$$\frac{x - mean(x)}{sd(x)}$$

Perform z-score scaling in R with the `scale` function:

```
x <- rnorm(50, mean = 50, sd = 10)
x_s <- scale(x, center = TRUE, scale = TRUE)
summary(x_s)
```

```
##        V1
##  Min.   :-2.36115
##  1st Qu.:-0.62046
##  Median :-0.05326
##  Mean   : 0.00000
##  3rd Qu.: 0.65266
##  Max.   : 2.53141
```

##### 15.1.3.4.5 Scaling variables

- Use `caret` package to scale multiple variables simultaneously and
    - get a reusable scaling model for applying to test data,
    - and eventually production data.

```
transformations <- preProcess(training_x)
scaledVars <- predict(transformations, training_x)
knitr::kable(t(summary(scaledVars)))
```

| | | | | | | |
|---|---|---|---|---|---|---|
| mpg | Min. :-1.57103 | 1st Qu.:-0.78724 | Median :-0.09845 | Mean : 0.00000 | 3rd Qu.: 0.51909 | Max. : 2.15002 |
| cyl | Min. :-1.106 | 1st Qu.:-1.106 | Median : 0.000 | Mean : 0.000 | 3rd Qu.: 1.106 | Max. : 1.106 |
| disp | Min. :-1.2010 | 1st Qu.:-0.8254 | Median :-0.4695 | Mean : 0.0000 | 3rd Qu.: 0.7957 | Max. : 1.8380 |
| hp | Min. :-1.2558 | 1st Qu.:-0.6895 | Median :-0.4333 | Mean : 0.0000 | 3rd Qu.: 0.6050 | Max. : 2.5603 |
| drat | Min. :-1.5108 | 1st Qu.:-0.8329 | Median : 0.1058 | Mean : 0.0000 | 3rd Qu.: 0.6447 | Max. : 2.2614 |
| wt | Min. :-1.52343 | 1st Qu.:-0.63886 | Median : 0.02664 | Mean : 0.00000 | 3rd Qu.: 0.30394 | Max. : 2.09156 |
| qsec | Min. :-1.72302 | 1st Qu.:-0.55308 | Median :-0.02315 | Mean : 0.00000 | 3rd Qu.: 0.52982 | Max. : 2.57785 |
| am | Min. :-0.9364 | 1st Qu.:-0.9364 | Median :-0.9364 | Mean : 0.0000 | 3rd Qu.: 1.0215 | Max. : 1.0215 |
| gear | Min. :-1.0623 | 1st Qu.:-1.0623 | Median : 0.2236 | Mean : 0.0000 | 3rd Qu.: 0.2236 | Max. : 1.5096 |
| carb | Min. :-1.0289 | 1st Qu.:-0.4654 | Median :-0.4654 | Mean : 0.0000 | 3rd Qu.: 0.6614 | Max. : 2.9151 |

#### 15.1.3.4.6 Things to check for

- Correlated variables
- Low variance columns

the `caret` package is very useful for these

#### 15.1.3.4.7 Missingness: How to handling missing values

- Common methods for coping with missing data:
  - Removing rows with missing values
    - Con: reduces sample size
    - Pro: use only complete data
  - [Continuous variables only] Putting in a default value like mean
    - Con: tends to flatten model coefficient for variable
    - Pro: simple to do
  - Putting in a predicted value
    - Con: requires another set of data
    - Pro: realistic values
  - [Continuous variables only] Making variable a categorical with an explicit missing category
    - Con: information loss on continuous variables
    - Pro: explicit modeling of missingness

### 15.1.3.5 Building models

#### 15.1.3.5.1 Initial models

- I try to build some candidate models:
  - All variables
  - A few strongest variables

```
fullmodel <- glm(training_y ~ .,
                 data = training_x,
                 family = binomial(link = "logit"))
steppedmodel <- step(fullmodel, direction = "both", trace = FALSE)
```

```
summary(steppedmodel)
```

#### 15.1.3.5.2 Stepwise variable selection

```
##
## Call:
## glm(formula = training_y ~ mpg + qsec, family = binomial(link = "logit"),
##     data = training_x)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.020e-05  -2.110e-08  -2.110e-08   2.110e-08   2.714e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1033.391 973364.150  -0.001    0.999
## mpg              7.609   8028.474   0.001    0.999
```

```
## qsec          48.745  47742.325   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.1841e+01  on 22  degrees of freedom
## Residual deviance: 1.7960e-09  on 20  degrees of freedom
## AIC: 6
##
## Number of Fisher Scoring iterations: 25
```

##### 15.1.3.5.3 Other model types

- Different logistic regression variants
  - like the `glmnet`, `glm` packages
- Different models
  - like classification trees

##### 15.1.3.5.4 Others

- You can also try with different loss or error functions
- You should try "common sense" models

#### 15.1.3.6 Evaluating `glms`

##### 15.1.3.6.1 `broom`

- Use `broom` to make tidy versions of model outputs.

```
library(broom)
# Coefficients
knitr::kable(tidy(steppedmodel))
```

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | -1033.390667 | 973364.150 | -0.0010617 | 0.9991529 |
| mpg | 7.609363 | 8028.474 | 0.0009478 | 0.9992438 |
| qsec | 48.744843 | 47742.325 | 0.0010210 | 0.9991854 |

##### 15.1.3.6.2 `broom`

- Use `broom` to make tidy versions of model outputs.

```
# Fitted data
knitr::kable(head(augment(steppedmodel)))
```

| .rownames | training_y | mpg | qsec | .fitted | .resid | .std.resid | .hat | .sigma | .cooksd |
|---|---|---|---|---|---|---|---|---|---|
| Mazda RX4 | 0 | 21.0 | 16.46 | -71.25393 | 0 | 0 | 1.1e-06 | 9.7e-06 | 0 |
| Datsun 710 | 1 | 22.8 | 18.61 | 47.24433 | 0 | 0 | 9.0e-07 | 9.7e-06 | 0 |
| Hornet 4 Drive | 1 | 21.4 | 19.44 | 77.04944 | 0 | 0 | 2.0e-06 | 9.7e-06 | 0 |
| Hornet Sportabout | 0 | 18.7 | 17.02 | -61.45836 | 0 | 0 | 1.1e-06 | 9.7e-06 | 0 |
| Valiant | 1 | 18.1 | 20.22 | 89.95952 | 0 | 0 | 3.2e-06 | 9.7e-06 | 0 |
| Duster 360 | 0 | 14.3 | 15.84 | -152.45847 | 0 | 0 | 4.8e-06 | 9.7e-06 | 0 |

#### 15.1.3.6.3 `broom`

- Use `broom` to make tidy versions of model outputs.

```
# Key statistics
knitr::kable(glance(steppedmodel))
```

| null.deviance | df.null | logLik | AIC | BIC | deviance | df.residual | nobs |
|---:|---:|---:|---:|---:|---:|---:|---:|
| 31.84128 | 22 | 0 | 6 | 9.406483 | 0 | 20 | 23 |

#### 15.1.3.6.4 Coefficients

- Are the coefficient signs in the right directions?
- How significant are they?
- How important are they?

#### 15.1.3.6.5 Key metrics

- *Residual deviance* is a measure of how much error is in the model,
  - after considering all the variables in the model.
  - The smaller the residual deviance, the better.

deviance

```
deviance(fullmodel)
```

```
## [1] 3.650173e-10
```

*Akaike's information criterion (AIC)*

- is a measure of information captured by a model
  - and penalizes more variables over fewer variables.
- The smaller the AIC, the better.

AIC information theory

The Akaike information criterion (AIC)

- is an estimator of out-of-sample prediction error
  - and thereby relative quality of statistical models
  - for a given set of data.
- Given a collection of models for the data,
  - AIC estimates the quality of each model,
  - relative to each of the other models.
- Thus, AIC provides a means for model selection.

AIC is founded on information theory.

- When a statistical model is used
  - to represent the process that generated the data,
  - the representation will almost never be exact;
- so some information will be lost
  - by using the model to represent the process.
- AIC estimates the relative amount of information lost by a given model:
  - the less information a model loses,
  - the higher the quality of that model.

```
AIC(fullmodel)
```

```
## [1] 22
```

#### 15.1.3.6.6 Classification rates

- A Confusion Matrix
    - is a specific table layout that allows
        * visualization of the performance of an algorithm,
        * typically a supervised learning one.
    - Each row of the matrix represents
        * the instances in a predicted class
        * while each column represents the instances in an actual class.
    - The name stems from the fact that it makes it easy to see
        * if the system is confusing two classes
        * (i.e. commonly mislabeling one as another).

It is a special kind of contingency table,

- with two dimensions ("actual" and "predicted"),
    - and identical sets of "classes" in both dimensions
    - (each combination of dimension and class
    - is a variable in the contingency table).

A contingency table

- (also known as a **cross tabulation** or **crosstab**)
- is a type of table in a matrix format
    - that displays the (multivariate) frequency distribution of the variables.
- They are heavily used in
    - survey research, business intelligence, engineering, & scientific research.
- They provide a basic picture of
    - the interrelation between two variables
    - and can help find interactions between them.

Lets look at the confusion matrix

- On the **training** data
- And **predicting** on the training data

```
training_pred <-
  ifelse(predict(steppedmodel, training_x) > 0, "1", "0")
training_pred <- factor(training_pred)
training_y <- factor(training_y)
confusionMatrix(training_pred, training_y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 12  0
##          1  0 11
##
##                Accuracy : 1
##                  95% CI : (0.8518, 1)
##     No Information Rate : 0.5217
##     P-Value [Acc > NIR] : 3.173e-07
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
```

```
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5217
##          Detection Rate : 0.5217
##    Detection Prevalence : 0.5217
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

##### 15.1.3.6.7 Classification rates

- Now lets look at the confusion matrix
  - On the **testing** data
  - And **predicting** on the testing data

```r
testing_pred <- ifelse(predict(fullmodel, testing_x) > 0, "1", "0")
testing_pred <- factor(testing_pred)
testing_y <- factor(testing_y)
confusionMatrix(testing_pred, testing_y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 3 0
##          1 3 3
##
##                Accuracy : 0.6667
##                  95% CI : (0.2993, 0.9251)
##     No Information Rate : 0.6667
##     P-Value [Acc > NIR] : 0.6503
##
##                   Kappa : 0.4
##
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 0.5000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.5000
##              Prevalence : 0.6667
##          Detection Rate : 0.3333
##    Detection Prevalence : 0.3333
##       Balanced Accuracy : 0.7500
##
##        'Positive' Class : 0
##
```

#### 15.1.3.7 Links

- Steph Locke