Data Visualizations & Machine Learning

April 8, 2021

ankura

Theory of Data Graphics

ankura

Data Graphics

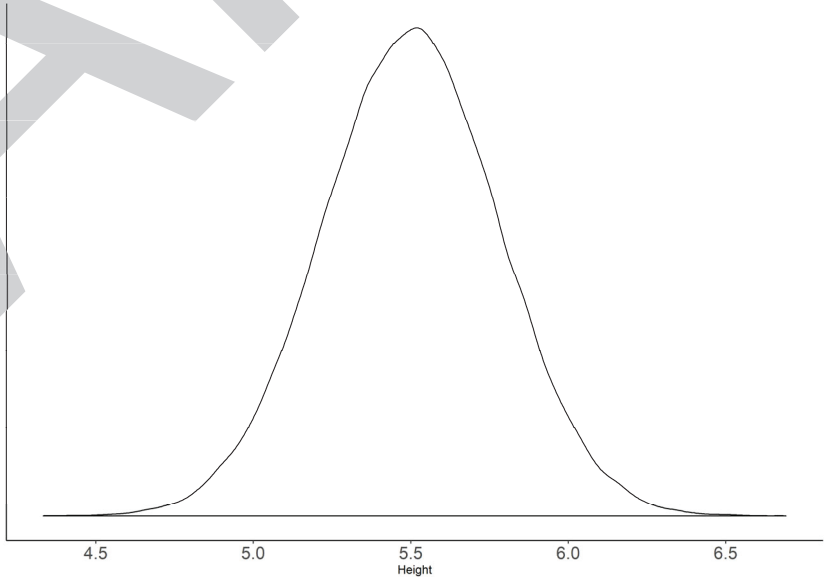
- Why Data Graphics?
  - Numbers are hard, pictures are easy.
  - What is the meaning behind all of these numbers?

5.4	5.6	4.7	5.7	5.2	5.8	5.1	5.5	6.2	5.6
5.8	5.8	5.4	5.2	5.4	5.3	5.7	5.1	5.4	5.7
5.2	5.8	5.6	5.0	5.3	5.9	5.3	5.6	5.6	5.7
5.5	5.5	5.2	5.5	5.6	5.7	5.7	5.6	5.2	5.0
5.8	5.0	5.6	5.6	5.7	5.5	5.4	5.3	6.0	5.7
5.2	5.5	5.3	5.6	5.2	5.4	5.8	5.0	5.3	5.7
5.6	5.8	5.5	5.0	5.3	5.4	5.4	5.6	6.0	5.6
5.1	6.0	5.4	5.8	5.5	5.6	5.6	6.0	5.5	5.2
5.5	5.9	5.5	5.6	6.2	5.6	5.8	5.7	5.6	5.0
5.0	5.6	5.3	6.1	6.2	5.6	5.4	5.5	5.1	5.4

Data Graphics

- Why Data Graphics?
  - Numbers are hard, pictures are easy.
  - What is the meaning behind all of these numbers?

Distribution of Employees' Heights at Ankura

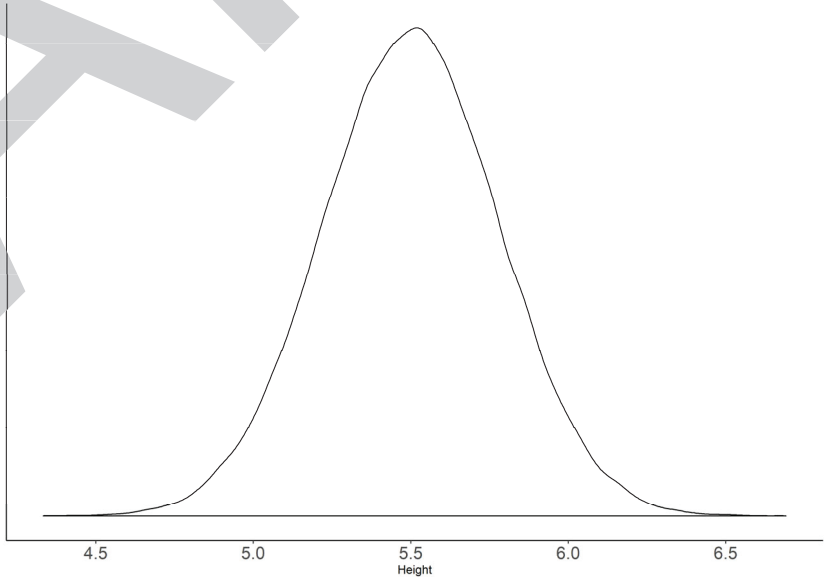


Data Graphics

- Why Data Graphics? **Because pictures have meaning where numbers may not.**

5.4	5.6	4.7	5.7	5.2	5.8	5.1	5.5	6.2	5.6
5.8	5.8	5.4	5.2	5.4	5.3	5.7	5.1	5.4	5.7
5.2	5.8	5.6	5.0	5.3	5.9	5.3	5.6	5.6	5.7
5.5	5.5	5.2	5.5	5.6	5.7	5.7	5.6	5.2	5.0
5.8	5.0	5.6	5.6	5.7	5.5	5.4	5.3	6.0	5.7
5.2	5.5	5.3	5.6	5.2	5.4	5.8	5.0	5.3	5.7
5.6	5.8	5.5	5.0	5.3	5.4	5.4	5.6	6.0	5.6
5.1	6.0	5.4	5.8	5.5	5.6	5.6	6.0	5.5	5.2
5.5	5.9	5.5	5.6	6.2	5.6	5.8	5.7	5.6	5.0
5.0	5.6	5.3	6.1	6.2	5.6	5.4	5.5	5.1	5.4

Distribution of Employees' Heights at Ankura



Data Graphics

- Let's take it a step further.

5.4	5.6	4.7	5.7	5.2	5.8	5.1	5.5	6.2	5.6
5.8	5.8	5.4	5.2	5.4	5.3	5.7	5.1	5.4	5.7
5.2	5.8	5.6	5.0	5.3	5.9	5.3	5.6	5.6	5.7
5.5	5.5	5.2	5.5	5.6	5.7	5.7	5.6	5.2	5.0
5.8	5.0	5.6	5.6	5.7	5.5	5.4	5.3	6.0	5.7
5.2	5.5	5.3	5.6	5.2	5.4	5.8	5.0	5.3	5.7
5.6	5.8	5.5	5.0	5.3	5.4	5.4	5.6	6.0	5.6
5.1	6.0	5.4	5.8	5.5	5.6	5.6	6.0	5.5	5.2
5.5	5.9	5.5	5.6	6.2	5.6	5.8	5.7	5.6	5.0
5.0	5.6	5.3	6.1	6.2	5.6	5.4	5.5	5.1	5.4

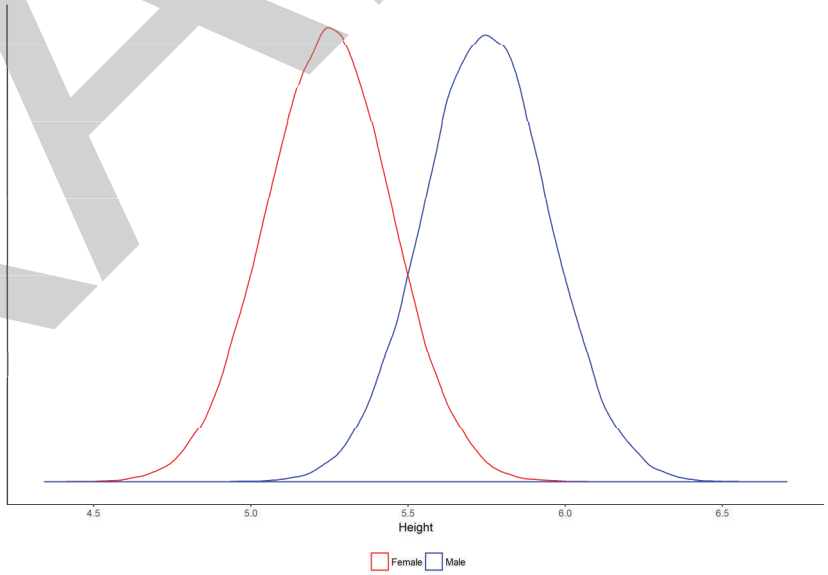
Let's color code men vs. women's heights.

With additional information, we can plot the data again, and this time tell a different story.

Data Graphics

- Let's take it a step further.

On Average, Men are Taller than Women at Ankura



Our story has a simple conclusion:

- On average, Men are taller than Women.
- However, some Women will be Taller than Men.

Data Graphics

- What is the moral of this story?
  - Data is so large that analyzing and interpreting it becomes very difficult.
  - Translating large amounts of data into a graph is a good way of highlighting a pattern or a trend.
  - Analyzing a lot of data is pretty useless, unless you can tell a good story from it.
  - **Graphs help elucidate that story.**

Data Graphics

- A few things to remember before we go on:
  - Long, cumbersome tables may convey lots of information, but often do not **tell the story in a simple and meaningful way**.
  - Graphs have **high data density**. This means that more information can be conveyed with less space than with a table or text.
  - Graphs can **highlight a pattern or a trend** better than text or a table.

Data Graphics

- **The Four Os**

Observable – The graphic contains a fact or a trend that a lay person can see. Visuals should speak for themselves.

Objective – The graphic does not attempt to hide a fact or trend, nor does it attempt to create one that is not there.

Original – The graphic contains verifiable data sources

Open – The graphic is clear and concise.
Simple is best!

Data Graphics

- **The Four Cs (sort of)**

Title – A simple but descriptive title that explains the graph.

Citations – The graph should include ALL data sources and citations.

Caption – If the title is not enough to explain what is happening, then a caption should be added.

Colors – Proper use of color is key to helping tell the story. Balance is key.

Building a Great Visual

Building a Great Visual

- Let's Work through a Hypothetical

Suppose I have the following summary table that I want to turn into a graphic.

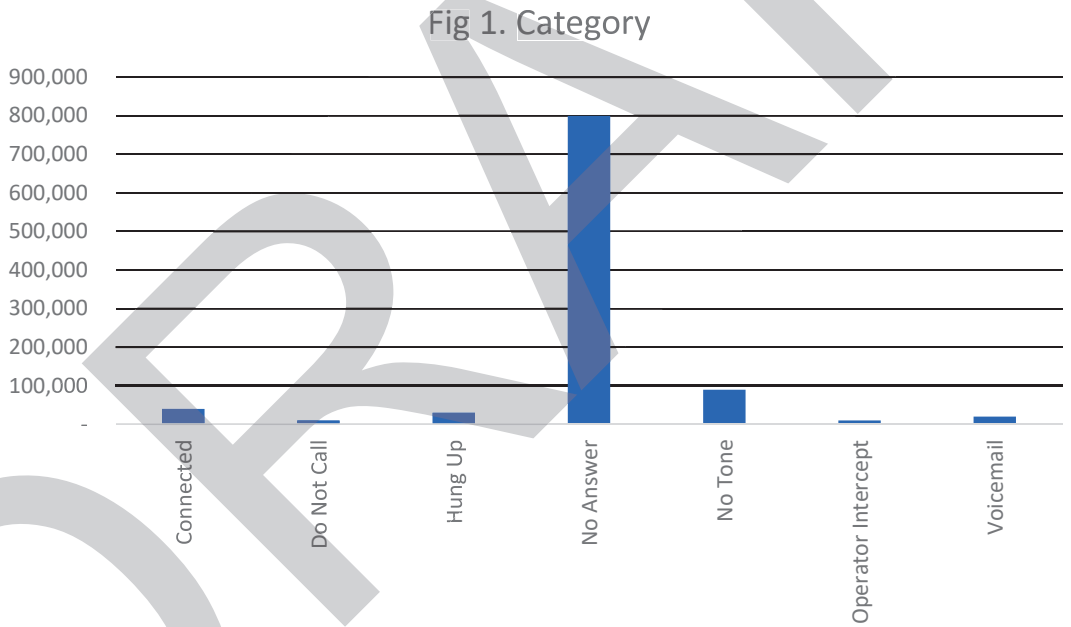
Table 1: Number of Calls by Category

Category	Number of Calls	% of Total
Do Not Call	10,000	1%
Operator Intercept	10,000	1%
Voicemail	20,000	2%
Hung Up	30,000	3%
Connected	40,000	4%
No Tone	90,000	9%
No Answer	800,000	80%
Total	1,000,000	

Building a Great Visual

- Let's Work through a Hypothetical

This is the base graphic you would get if you used Excel. It is not very good.

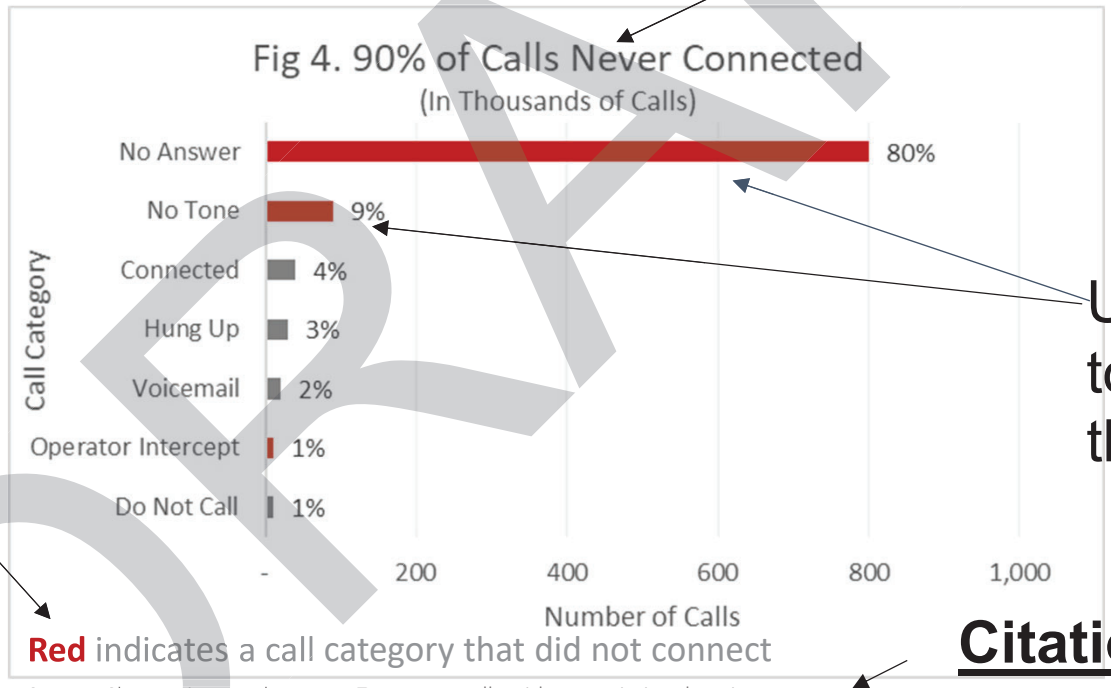


Building a Great Visual

- Let's Work through a Hypothetical

This graphic tells a story via the image.

Descriptive **Title**



Source: Shoutpoint Database, 2017 to 2018, Calls with non-missing durations.

Use of **Color** to highlight the story

Caption

Citation to sources

Building a Great Visual

How do we know what type of graph to choose?

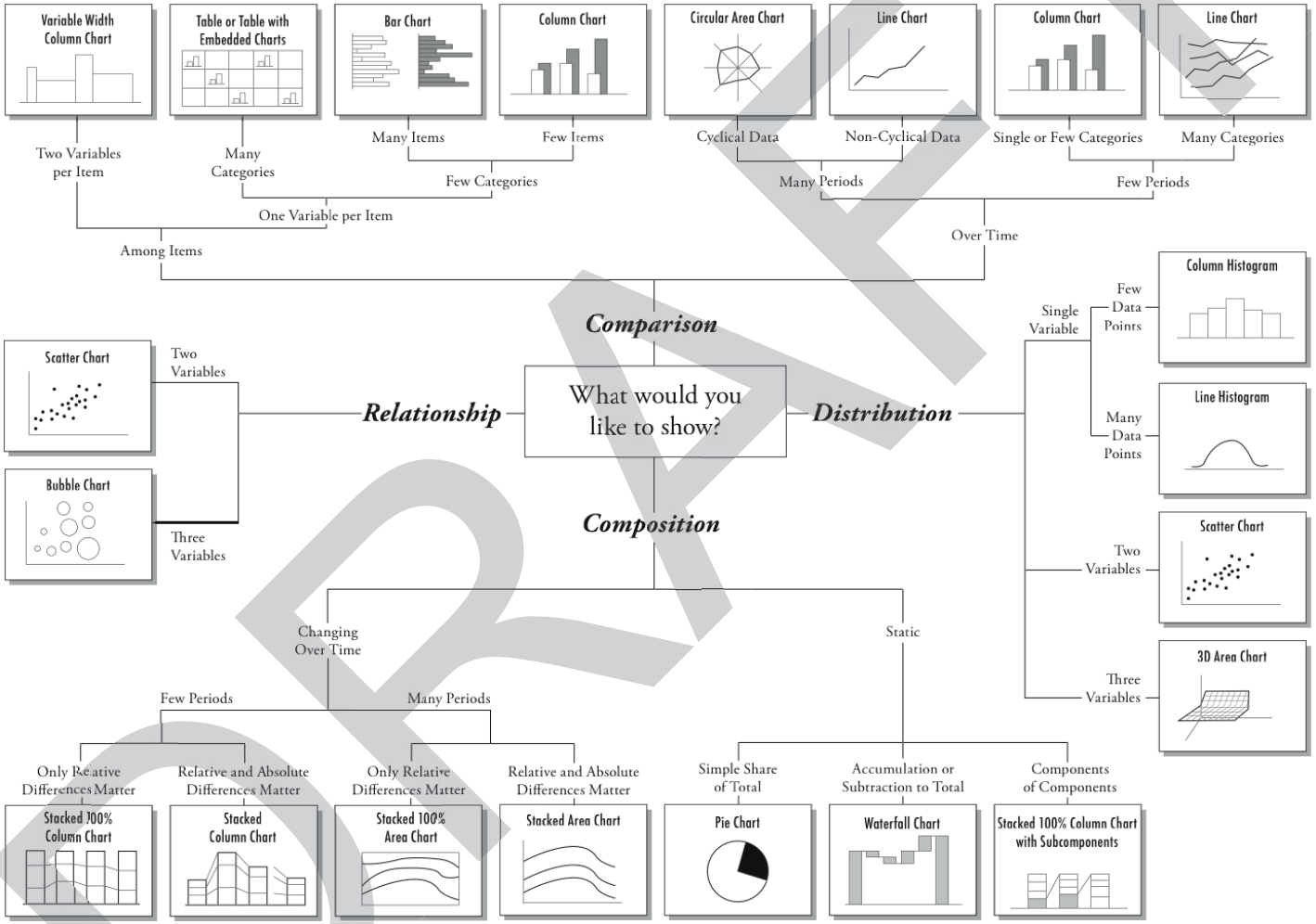
- So far, I have only presented column graphs.
- However, different types of data dictate different types of graphs.
- Some simple rules:
  1. Categorical Data – Column / Bar Graph
  2. Continuous Data – Histograms, Density Graph
  3. Data that exists over time – Timelines
  4. Relationships – Scatterplots

Building a Great Visual

How do we know what type of graph to choose?

- Beyond just your data, what you would like to show with your data dictates the types of graphs you should use.
- Some additional considerations:
    1. Comparison
    2. Relationship
    3. Composition
    4. Distribution

Building a Great Visual



Building a Great Visual

Summary

- Visuals help tell the story better than words or tables
- Be wary of putting too much information into a graph. **Simple is best!**
- The four Os can help guide you to design a good graph: **Objective, Observable, Original and Open**.
- The four Cs (sort of) make your graph complete and help it stand-alone: **Title, Caption, Citation, Color**.

ggplot() Basics

ggplot() basics

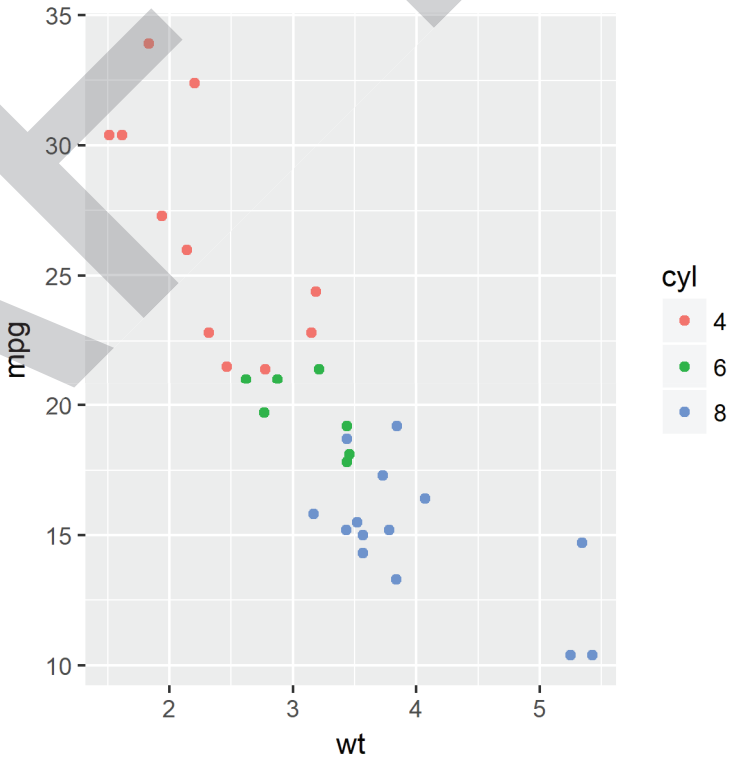
ggplot() is the go-to R package for graphics.

- It operates on the concept of the Grammar of Graphics
- The Grammar of Graphics follows the following format
  - Data
  - Aesthetics
  - Geom
  - Statistics / Transformations

ggplot() basics

ggplot() in detail

- "ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms** – visual marks that represent data points, and a **coordinate system**."
- Data and Geoms are user-defined, and the default coordinate system is a grid.
- Users can define alternative coordinate systems—such as a polar coordinate system to wrap their data in a circle around a central point.
- In addition to these three essentials, we will also consider **aesthetics** and **themes** as important features for visualizations.




```
p ← ggplot(data = cars, aes(x = wt, y = mpg, colour = as.factor(cyl)))
p ← p+geom_point()
p
```

ggplot() basics

ggplot() is the go-to R package for graphics.

- data – Your dataset, in this case “dta”

  ggplot(dta, ...

- aesthetics – Your variables, in this case “var1” and “var2”

  ggplot(dta, aes(x=var1, y=var2))...

- geoms – The shape of your graph (there are MANY!)

  ggplot(dta, aes(x=var1, y=var2))+geom_line()

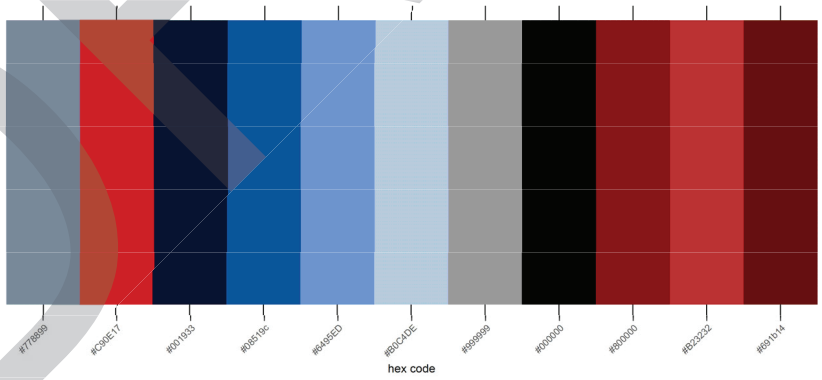
- statistics / transformations – Transformations to the data

  (Not Recommended, because you should transform your data before plotting)

ggplot() basics

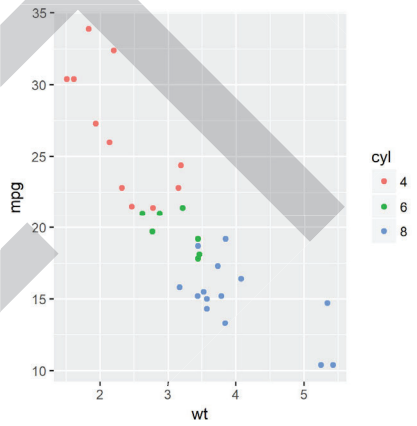
ggplot() is modular

- You can develop a standardized color scheme and general theme to help quickly format plots. You should do this because the standard colors for ggplot and color themes are not appealing.
- The reds, blues, and grey-scale colors can generally be used together, the order of the color-scheme vector is flexible.
  - Color use should be intentional – quickly highlighting for your audience your most important points, or simply making it easier to navigate your visual.
  - out_theme <- theme_bw() + theme(text=element_text(family="ArialMT"), legend.position="bottom", plot.title = element_text(size = rel(2)), axis.text.x = element_text(size= rel(1)), axis.text.y = element_text(size= rel(1)))
  - color_scheme <- c("#778899", "#C90E17", "#001933", "#08519c", "#6495ED", "#B0C4DE", "#999999", "#000000", "#800000", "#B23232", "#691b14")

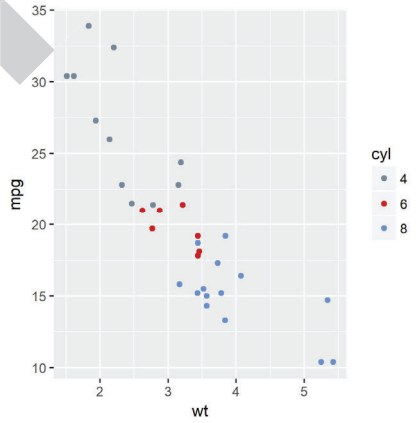


ggplot() basics

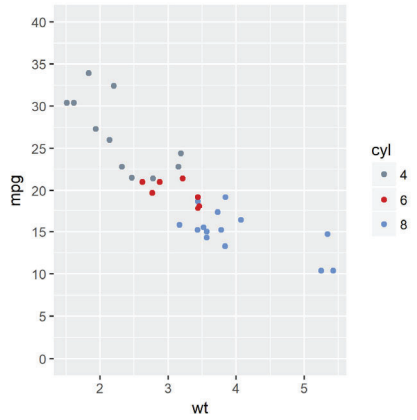
```
cars$cyl ← factor(cars$cyl, levels = sort(unique(cars$cyl)), ordered = TRUE)
cars$disp_size ← with(cars, ifelse(disp<=150, 'Small',
                                   ifelse(disp>120 & disp<=300, 'Medium', 'Large')))

p ← ggplot(data = cars, aes(x = wt, y = mpg, colour = cyl)) +geom_point()
```




```
p ← p + scale_color_manual(values = color_scheme[c(1,2,5)])
```

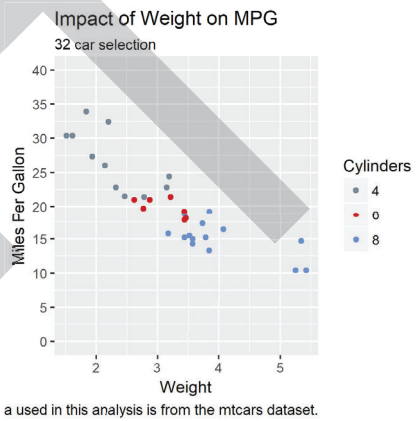



```
p ← p + scale_y_continuous(limits = c(0,40), breaks = seq(0,40,5))
```

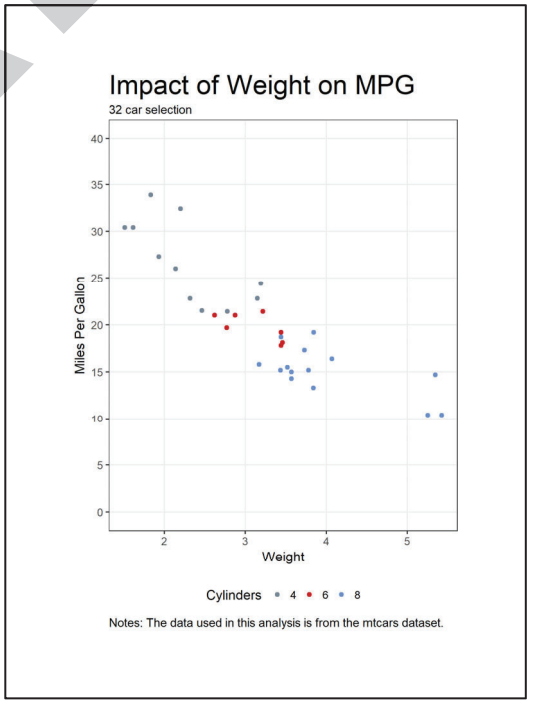


ggplot() basics

p ← p + labs(title="Impact of Weight on MPG", subtitle = "32 car selection", x = "Weight", y = "Miles Per Gallon", color = "Cylinders", caption = "Notes: The data used in this analysis is from the mtcars dataset.")



p ← p + out_theme + theme(panel.grid.minor = element_blank(), plot.caption = element_text(hjust=0), plot.margin=unit(c(2,2,2,2),"cm"))

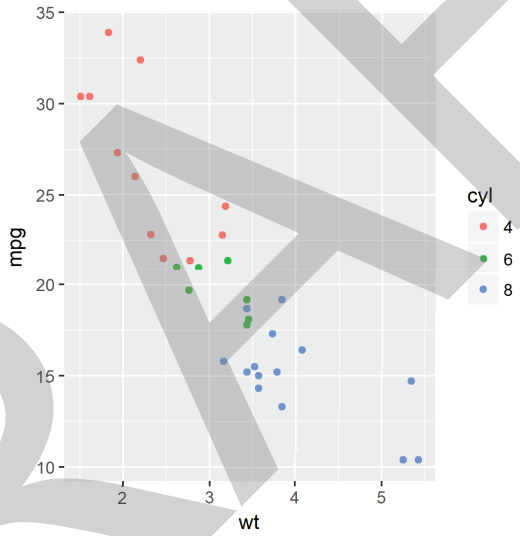


ggplot() basics

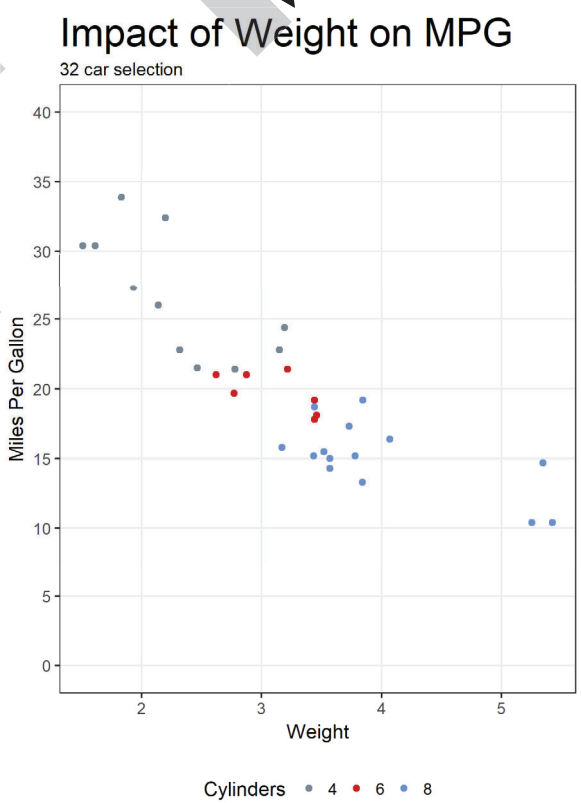
The Result:

```
p ← ggplot(data = cars, aes(x = wt, y = mpg))
p ← p + geom_point(aes(colour = as.factor(cyl)))

p ← p + scale_color_manual(values =
color_scheme[c(1,2,5)])

p ← p + scale_y_continuous(limits = c(0,40),
                           breaks = seq(0,40,5))
p ← p + labs(title="Impact of Weight on MPG",
                    subtitle = "32 car selection",
                    x = "Weight",
                    y = "Miles Per Gallon",
                   color = "Cylinders",
                    caption = "Notes: The data used in
this analysis is from the mtcars dataset ")

p ← p + out_theme
p ← p + theme(panel.grid.minor = element_blank(),
              plot.caption = element_text(hjust=0),
              plot.margin=unit(c(2,2,2,2),"cm"))
```

Start



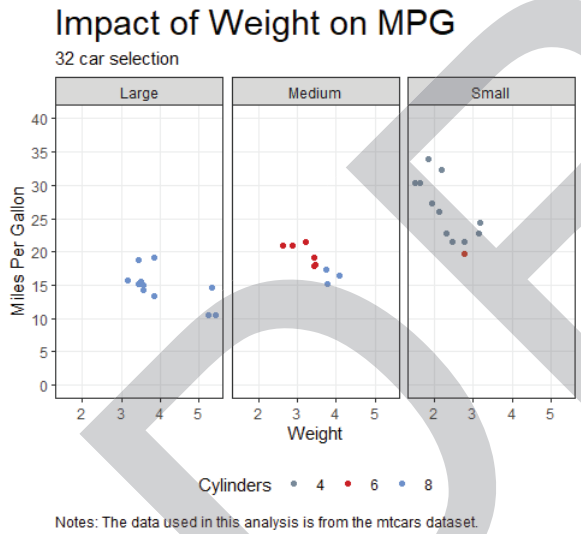
End



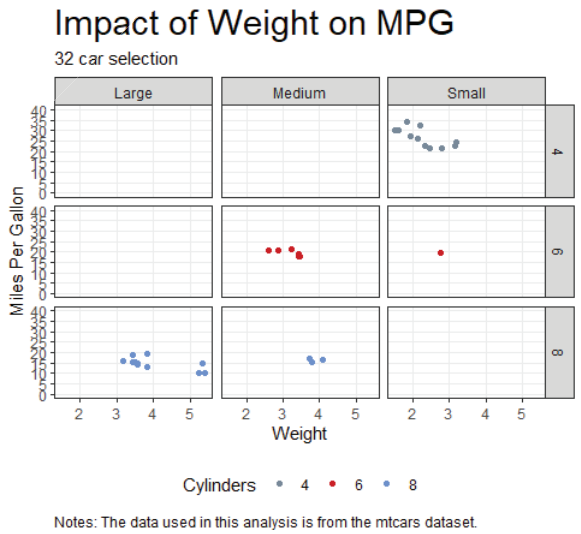
Faceting()

- Great way to split your data for exploration.
- Relies on categorical features.
- You can use facet_wrap() or facet_grid() to split up the data.

p ← p + facet(~disp_size)

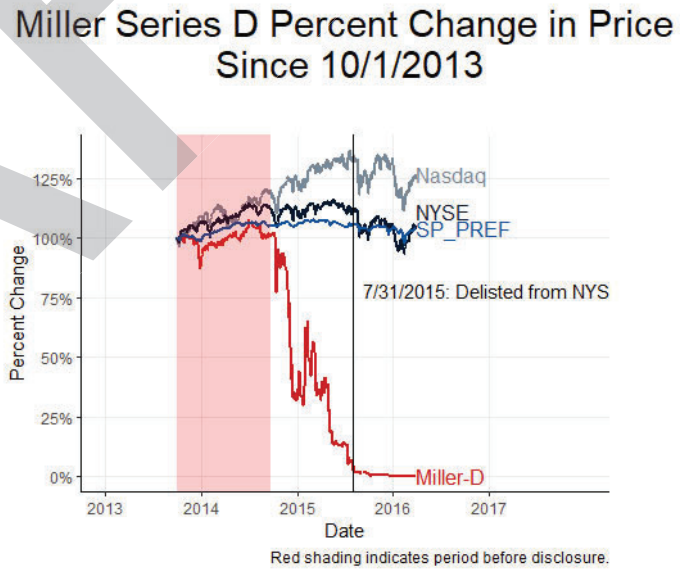


p ← p + facet(cyl~disp_size)



Another example (line plot)

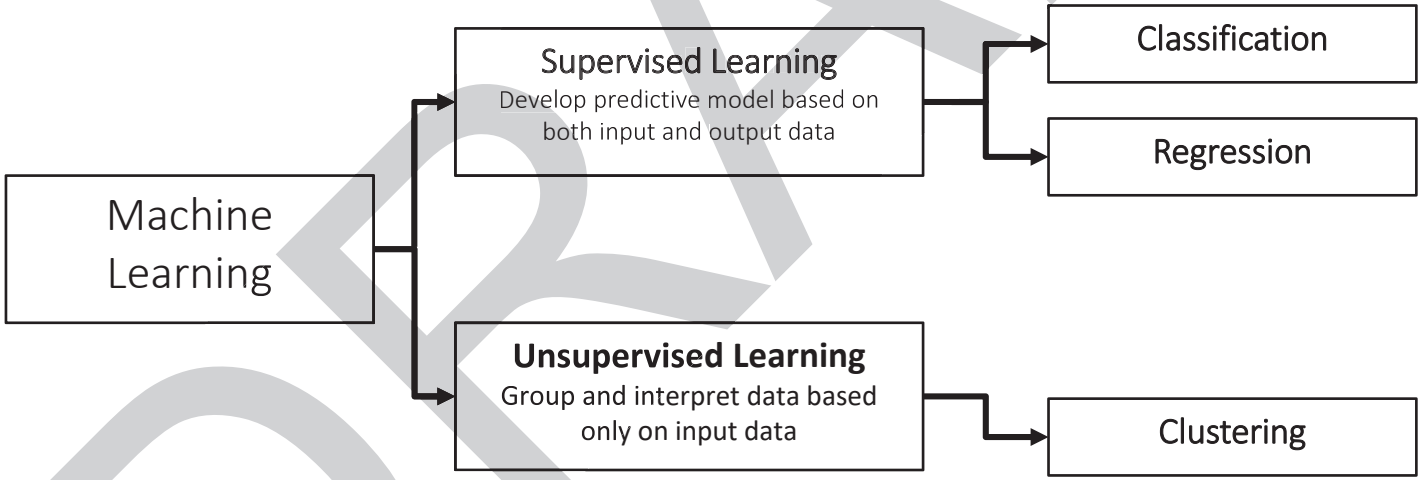
```
p2 <- ggplot(data = miller, aes(x=date, y = pct_change, colour = ticker))
p2 <- p2+geom_line(size=1)
p2 <- p2+scale_color_manual(values=color_scheme[c(2,1,3,4)])
p2 <- p2+geom_dl(aes(label = ticker), method = list(dl.combine("last.points")
        , cex = 1,"last.bumpup"))
p2 <- p2+scale_x_date(limits=as.Date(c("2013-01-01","2018-01-01"))
             ,breaks = seq.Date(as.Date("2013-01-01")
             ,as.Date("2017-01-01") , by = "1 year"),  date_labels = "%Y")
p2 <- p2+scale_y_continuous(labels=scales::percent,
                    breaks = seq(0,2,.25))
p2 <- p2+ out_theme
p2 <- p2 + theme(panel.grid.minor = element_blank(),
                 plot.title = element_text(hjust=.5),
                 plot.margin=unit(c(2,2,2,2),"cm"))
p2 <- p2 + labs(title="Miller Series D Percent
                          Change in Price\n Since 10/1/2013\n",
           x="Date",
           y="Percent Change",
           color="Ticker:  ")
p2 <- p2+geom_vline(xintercept = as.Date("2015-07-31"))
p2 <- p2+annotate("text",x = as.Date("2015-07-31"), y = .75
        , label = "  7/31/2015: Delisted from NYSE", hjust = 0, vjust=0
        , angle=0)
p2 <- p2+theme(legend.position = "none",
           panel.border = element_blank(),
           axis.line = element_line(color = "black"))
p2 <- p2+annotate("rect",xmin=as.Date("2013-10-01"), xmax=as.Date("2014-09-
22"), ymin=-Inf, ymax=Inf, fill="#FF000033")
```



Introduction to Machine Learning

Machine Learning Basics

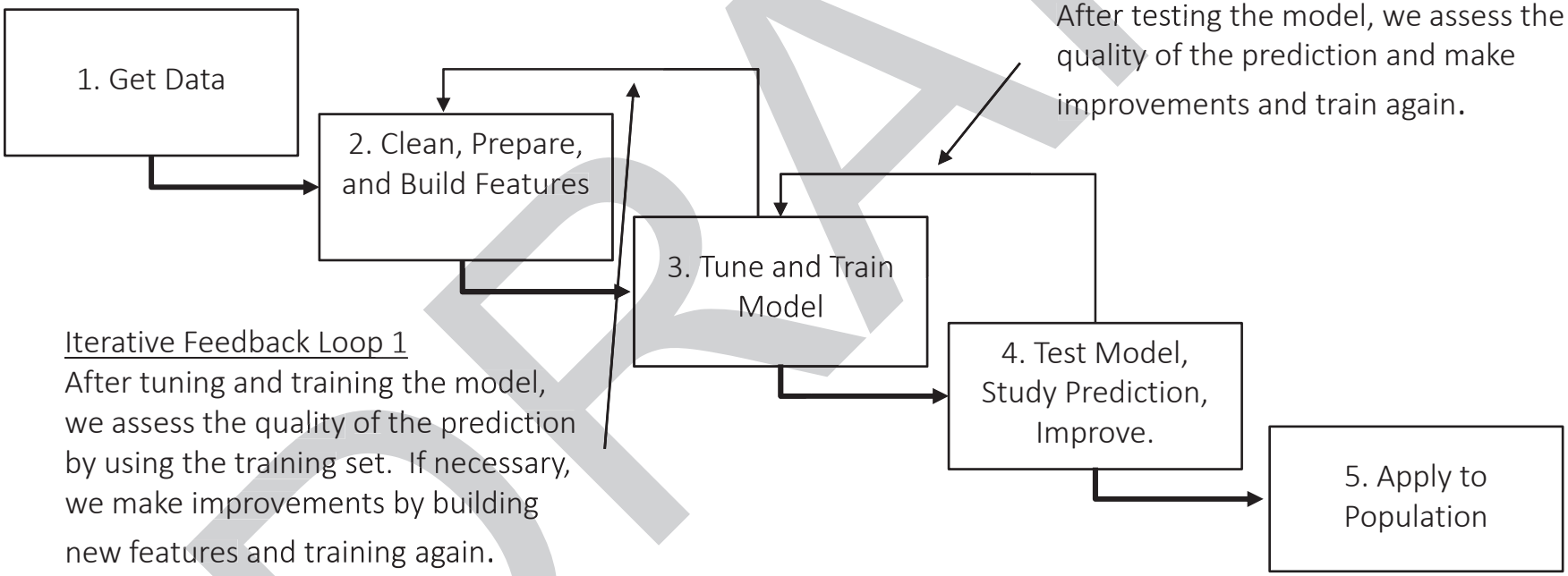
- We are going to discuss a couple of terms that are applicable to Machine Learning and what we performed. There are many more terms and processes under the Machine Learning umbrella that we do not have the time to cover today.
- Today we will focus on binary classification, which is a method for developing a model that predicts the output as either Match or No Match.



Machine Learning Basics

Supervised Learning - Classification:

Five "easy" steps



Iterative Feedback Loop 2
After testing the model, we assess the quality of the prediction and make improvements and train again.

Iterative Feedback Loop 1
After tuning and training the model, we assess the quality of the prediction by using the training set. If necessary, we make improvements by building new features and training again.

Machine Learning Basics

Supervised Learning - Classification:

Step 1: Get Data

1. Get Data

Machine Learning is data hungry. We need "a lot" of data to create a model that makes accurate predictions. Why?

a. The underlying algorithms make predictions, assess the error of those predictions, and then adjust to improve the prediction. This is done automatically and iteratively.
b. The more data that is fed into the model, the more accurately the model will predict.

Machine Learning Basics

Supervised Learning - Classification:

Step 2: Clean, Prepare, and Build Features

2. Clean, Prepare, and Build Features.

What does it mean to to "Build Features?"

a. Most of the data we encounter is not ready for any level of sophisticated analysis.
b. For machine learning we need to develop features (i.e. variables) that will be the inputs to the model.
c. In this case, we took the names on each record and
   i. Standardized names (removed spaces, punctuation, etc…)
   ii. Developed string distance metrics and phonetic mapping (more on that later).
   iii. Developed other key variables: first initial, last initial, and length of first and last name.

Machine Learning Basics

Supervised Learning - Classification:

Step 3: Tune and Train the Model

1. Generate a random sample of records from the population.
2. Split the random sample into a "training set" and a "test set." The test set is typically much smaller than the training set.
3. Have a human review each record and assign the "answers" to the sample. (This is the "supervised" part of the process.)

3. Tune and Train Model

4. Use sample to prepare model.
5. Use processes to either randomly assign tuning parameters or select your own tuning parameters.
6. Run model only on the training set. Compute and assess training accuracy.
7. If training accuracy is acceptable (for the problem at hand), move on to testing. If not, build more features and continue tuning and training.

Machine Learning Basics

Supervised Learning - Classification:

Step 4: Test Model, Study Prediction, and Improve

1. Apply the model created in the training stage to the "test" set that was randomly drawn from the population.
2. Compare the predicted values from the model to actual values that a human assigned to the observations.
3. Calculate accuracy and other diagnostic assessments of the model.
4. If accuracy and other diagnostics are acceptable, move on and apply the model to the full population. Typically, this will take many tries.
5. If diagnostics and / or accuracy are not acceptable, go back to training and try again.

4. Test Model, Study Prediction, Improve.

Machine Learning Basics

Supervised Learning - Classification:

Step 5: Apply to Population

1. Ensure features developed during training and testing also exist for every record in the population.
2. Apply the model to the population.
3. Review data and test prediction against other parameters.

5. Apply to Population

Machine Learning Basics

Supervised Learning - Classification:

How is it useful?

- Speed: review a small set of records and the machine does the rest.
- Power: the "machine" identifies patterns based upon learning provided by the human.
- Accuracy: the "machine" can be more accurate than a human review. Once the "machine" learns the patterns within the data, it is unbiased to those patterns.

How is it not?

- It takes time to get the model correct.
- An incorrectly specified model will lead to bad results (i.e. garbage in, garbage out).
- YOU often lose the ability to properly explain WHY decisions were made a certain way (although this is becoming less true with technological advances).
- Accuracy: Once the "machine" learns the patterns within the data, it is unbiased to those patterns. If these patterns are not detected or not taught to the machine properly, then the prediction can and will be biased.

Machine Learning Basics

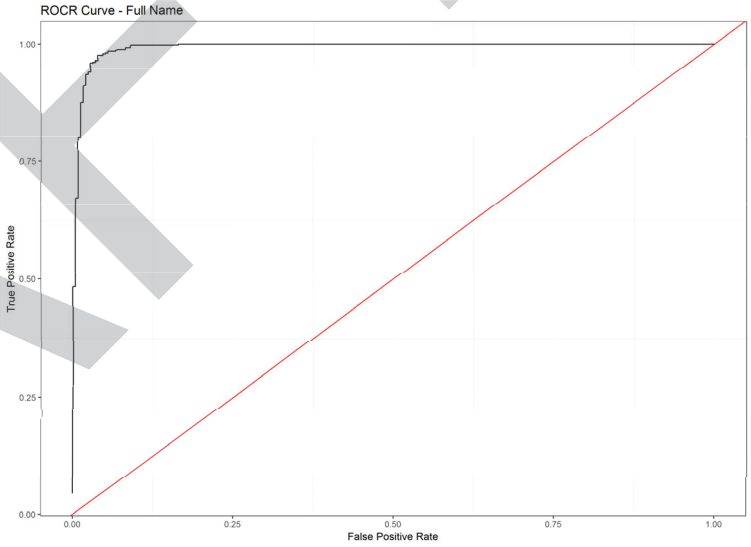
How to evaluate the models?

- In typical regression modeling, we would evaluate a model in a variety of ways:
  - Does the model fit the theory being tested?
  - Is the model significant?
  - Are the variables we are testing significant? Do they have a proper interpretation?
- We would also run diagnostics to ensure that proper modeling assumptions were maintained.
  - Residual plots
  - QQ Plots
  - Specific statistical tests to measure collinearity, auto-correlation, etc…
- We DO NOT need to evaluate these things in machine learning.
  - We evaluate the output and how accurate our prediction using ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve). These evaluation techniques are specific to binomial prediction.

Machine Learning Basics

How to evaluate the models?

- ROC Curve
  - Common technique to evaluate Machine Learning Binary Classifier models
  - The ROC also provides a single measurement of prediction accuracy called AUC (Area Under the Curve).
  - The process assigns different thresholds from 0.01 to 0.99 and then checks the classification at each threshold.



- Confusion Matrix
  - Shows all possible combinations of the prediction.
  - Easy to interpret and determine False Positives and True Negatives.
  - Makes determination based on a 50% threshold.

Test Set - Full Name Match Model

		True State			
		Match	No-match	Error	Rate
Predicated State	Match	62	7	10%	=7/69
	No-match	2	129	2%	=2/131
	Totals	64	136	5%	=9/200

Basic Example

Machine Learning Simple Example

Visuals, Story Telling, and Machine Learning

- We just spent a lot of time on how to craft a great visual. What does any of that have to do with Machine Learning?
  - Machine Learning is extremely obfuscated (i.e., it is a black box).
  - Visuals can help you tell your machine learning story effectively.
  - Visuals can help you explain WHY you are modeling the data in a specific way.
  - Visuals can help you explain the meaning of your Machine Learning output.
- We are going to work through an example involving binomial classification of data and using visuals to tell the story regarding the model.

Machine Learning Simple Example

UCI Credit Card Data

- We are trying to predict which people will default on their credit based on prior payment history and demographic information.
- The data was compiled by UCI and covers 30,000 unknown individuals with their credit card balances and payment history
- Data Sample

ID	LIMIT_BAL	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default.
1	20000	2	1	24	2	2	-1	-1	-2	-2	3913	3102	689	0	0	0	0	689	0	0	0	0	1
2	120000	2	2	26	-1	2	0	0	0	2	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0	2000	1
3	90000	2	2	34	0	0	0	0	0	0	29239	14027	13559	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
4	50000	2	1	37	0	0	0	0	0	0	46990	48233	49291	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
5	50000	2	1	57	-1	0	-1	0	0	0	8617	5670	35835	20940	19146	19131	2000	36681	10000	9000	689	679	0
6	50000	1	2	37	0	0	0	0	0	0	64400	57069	57608	19394	19620	20024	2500	1815	657	1000	1000	800	0

Machine Learning Simple Example

Step 1 – Gather Data (Done)

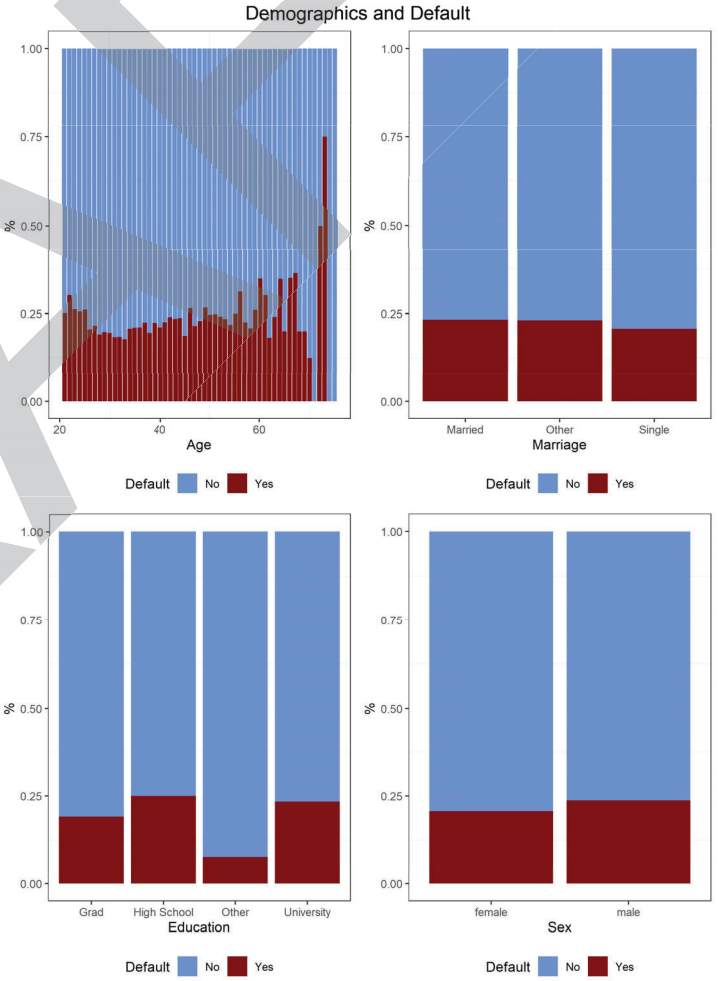
Step 2 – Clean, Prepare and Build Features (Why?)

- Recall that the data you are going to be analyzing is never clean, never ready for analysis, and likely needs new variables created (from existing ones and from other sources) to conduct a complete analysis.
- Based on the existing data here are some features I have thought of:
  - Average and Total Payment Status
  - Throughput of the account
  - Existing Balance
  - Etc…

Machine Learning Simple Example

Step 2 – Clean, Prepare and Build Features (Cont.)

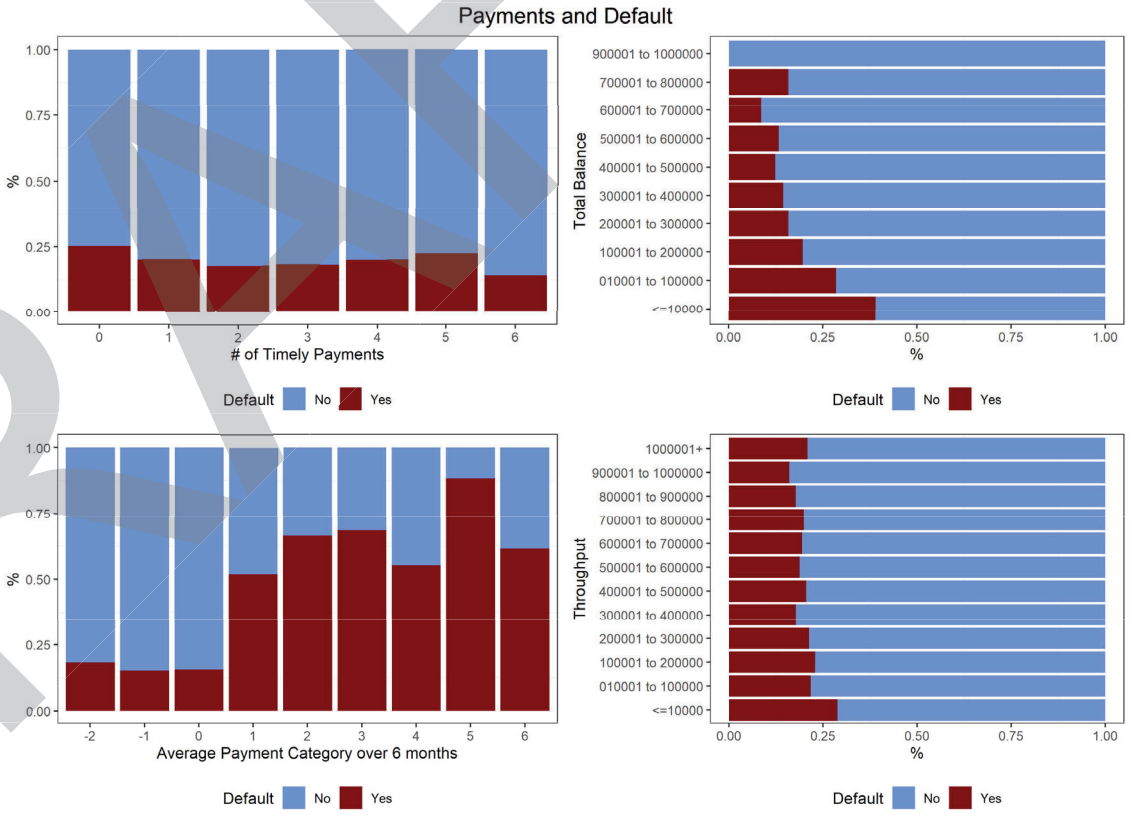
- Let's build some graphs to explore the data
- You can use the graphics to demonstrate which variables should be included in your model
- This helps you tell your story and helps you make decisions on which variables should be included.



Machine Learning Simple Example

Step 2 – Clean, Prepare and Build Features (Cont.)

- Here are some additional graphs
- Notice anything?
  - Payment Categories >0 have a significantly higher default rate
  - Balance Limits < 10,000 have higher default rates as well



Machine Learning Simple Example

Step 3 – Tune and Train the Model

- We are going to use a very basic glm() binomial model using R.
- There is not really any "tuning" of the model that we are going to be using.
- However, more advanced algorithms, such as neural nets, random forest, gradient boosted machines, etc… allow for very precise adjustments for modeling.
  - These adjustments are called tuning parameters.
  - They affect how the machine learns.
  - They will bias your results, and if you over-tune your model then you can overfit your training data. This will lead to poor results when you test your model.
- There is also cross-validation, where you take random samples of your training data and continually train and test samples on the training data to get "average" results. These results are typically better than a simple model.

Machine Learning Simple Example

Step 3 – Tune and Train the Model

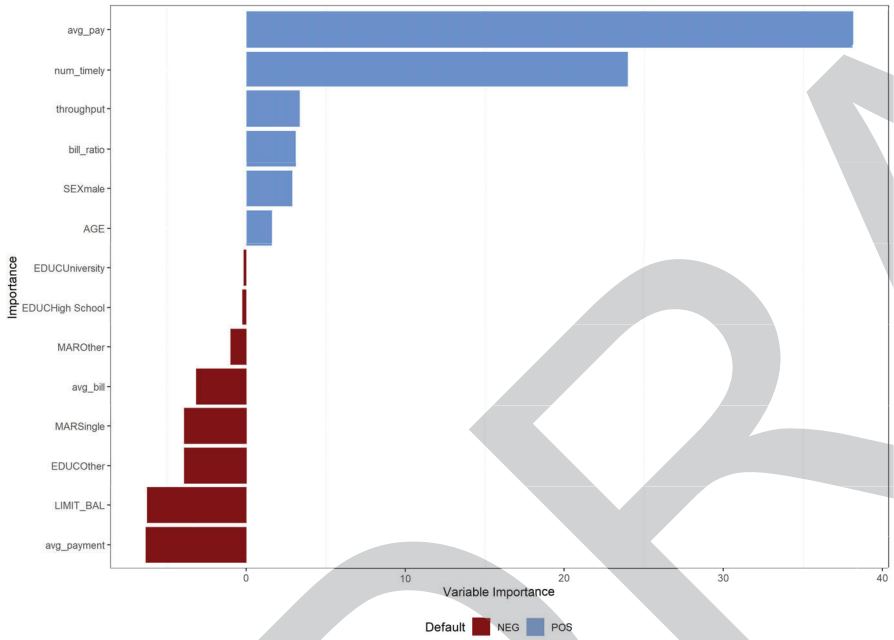
- Examples of Tuning Parameters
  - Cross Validation "Folds" - Number of times to cross-validate the model.
  - Alpha – Regularization.
  - Lambda – Strength of the regularization.
  - Lambda Search – Let the model find the optimal lambda value.
  - Early Stopping – Should the model continue if little iterative improvement occurs at each step or wait until convergence.
  - Standardized – Let the model automatically standardize the inputs.
- Randomized Grid Search
  - This process assigns a random value to each parameter above and runs the model.
  - This process is repeated with randomly assigned inputs for as+ many times as the analyst desires.
  - Accuracy is computed at each iteration of this process and is saved.
  - At the end of the process each of the models generated is ranked from highest accuracy to lowest accuracy. The model with the highest accuracy is selected as the "best" model.
  - The analyst loses control over selecting the tuning inputs, but if randomized grid search is allowed to perform 100s or 1000s of iterations, there should be enough models generated that the "best" model is found.

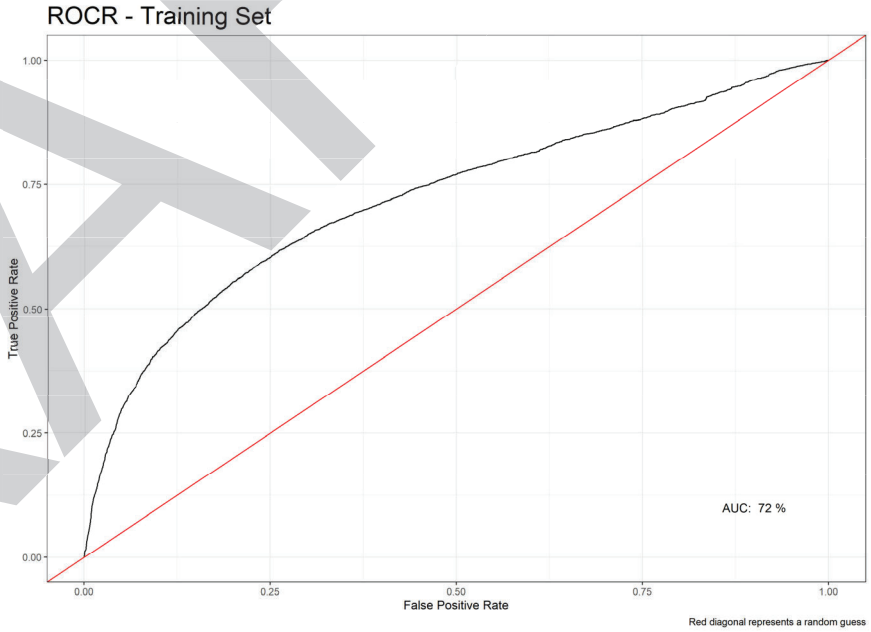
More here: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/grid-search.html

Machine Learning Simple Example

Step 3 – Tune and Train the Model (Cont.)

- You always want to evaluate training model accuracy and which variables are important.



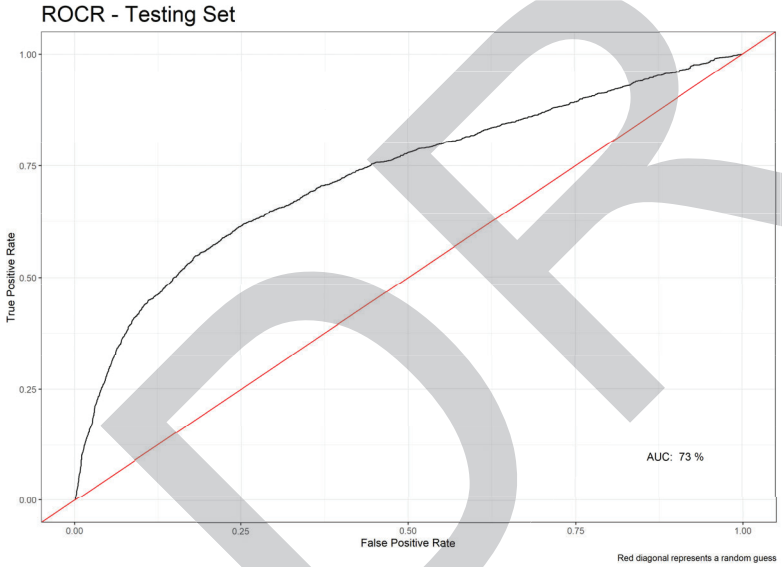


	Predict No Default	Predict Default	Error %	Total
No Default	15,915	481	3%	16,396
Default	3,635	969	79%	4,604
Total	19,550	1,450	20%	21,000

Machine Learning Simple Example

Step 4 / 5 – Test Model

- Many of our features did not have much importance (i.e., around 0)
  - I removed them and fitted a model with only Balance, Payment History, and the Number of Timely Payments.
  - The model is much simpler, which means it is easier to explain and implement. There is little change in the output results, which is good.
  - If the testing phase shows reasonable accuracy (compared to your expectations and the training accuracy), and you are happy with the theory underlying the approach, then you are done.



	Predict No Default	Predict Default	Error %	Total
No Default	6,758	210	3%	6,968
Default	1,606	426	79%	2,032
Total	8,364	636	20%	9,000

Machine Learning Simple Example

Other Considerations

- My model is not very good at predicting default. This is not surprising:
  - I used a very simple model. Even though it is technically "machine learning" more powerful algorithms would have given me a much more accurate results because those algorithms learn from their mistakes iteratively.
  - The sample is not balanced. There are many instances of "No Default" in the data compared to default.
    - One solution would have been to up-sample the "Default" results to balance the data.
    - If I had naively predicted everything as a "No Default," I still would have obtained fairly accurate predictions.
  - Binary is NOT really the best way to model this data. Scoring each person's risk of default is likely the better method.

Machine Learning Simple Example

Summary

- Separate into training / test sets.
- Explore the data and build Features. Features should be based on your theory and training data.
- Explore those Features. Graph everything!
- Develop your model.
- Tune it on your training set.
- Test it on your testing set.
- Validate your model against real world expectations and known information.

Questions?

Jeremy Guinta
jeremy.guinta@ankura.com