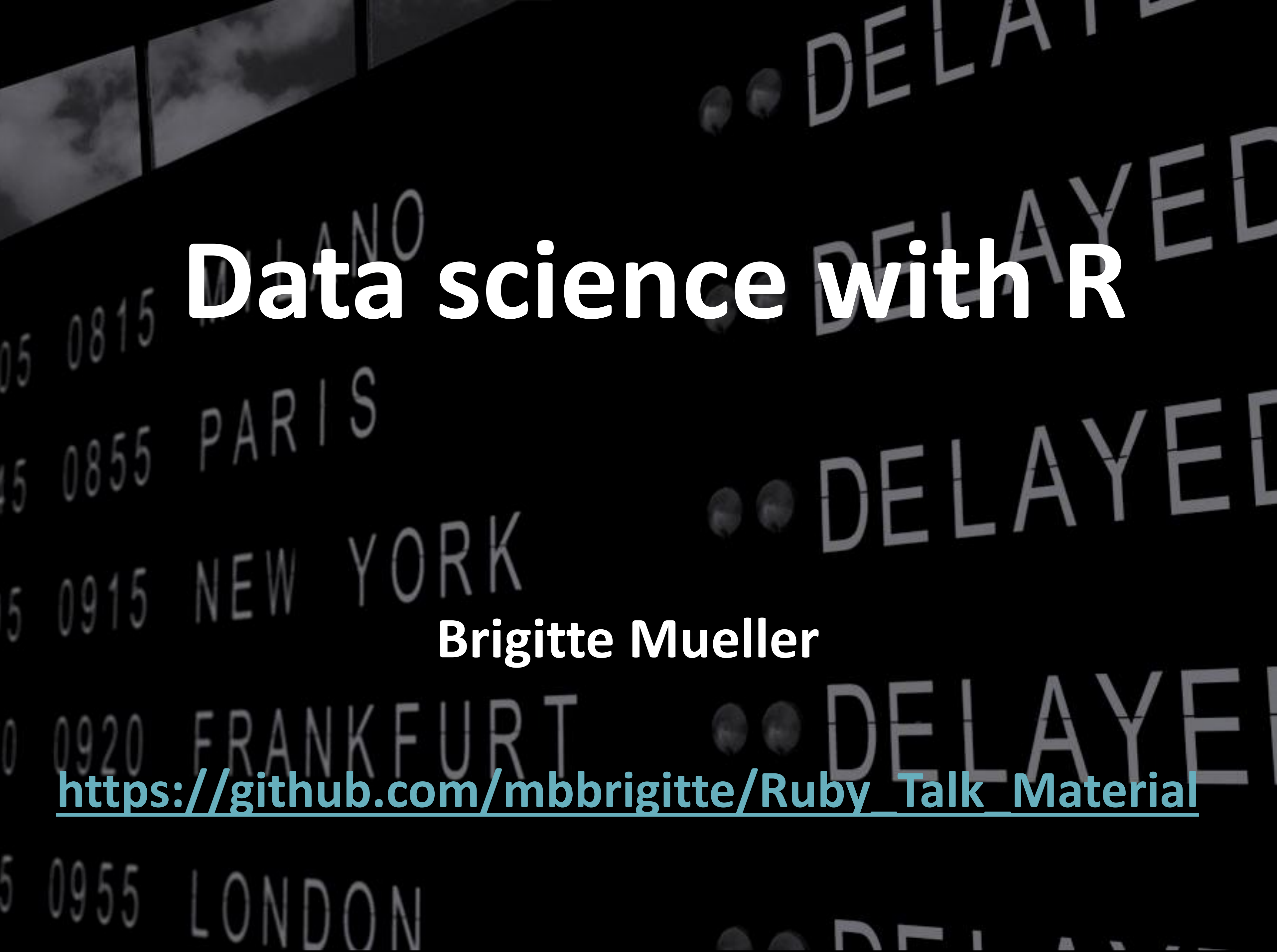
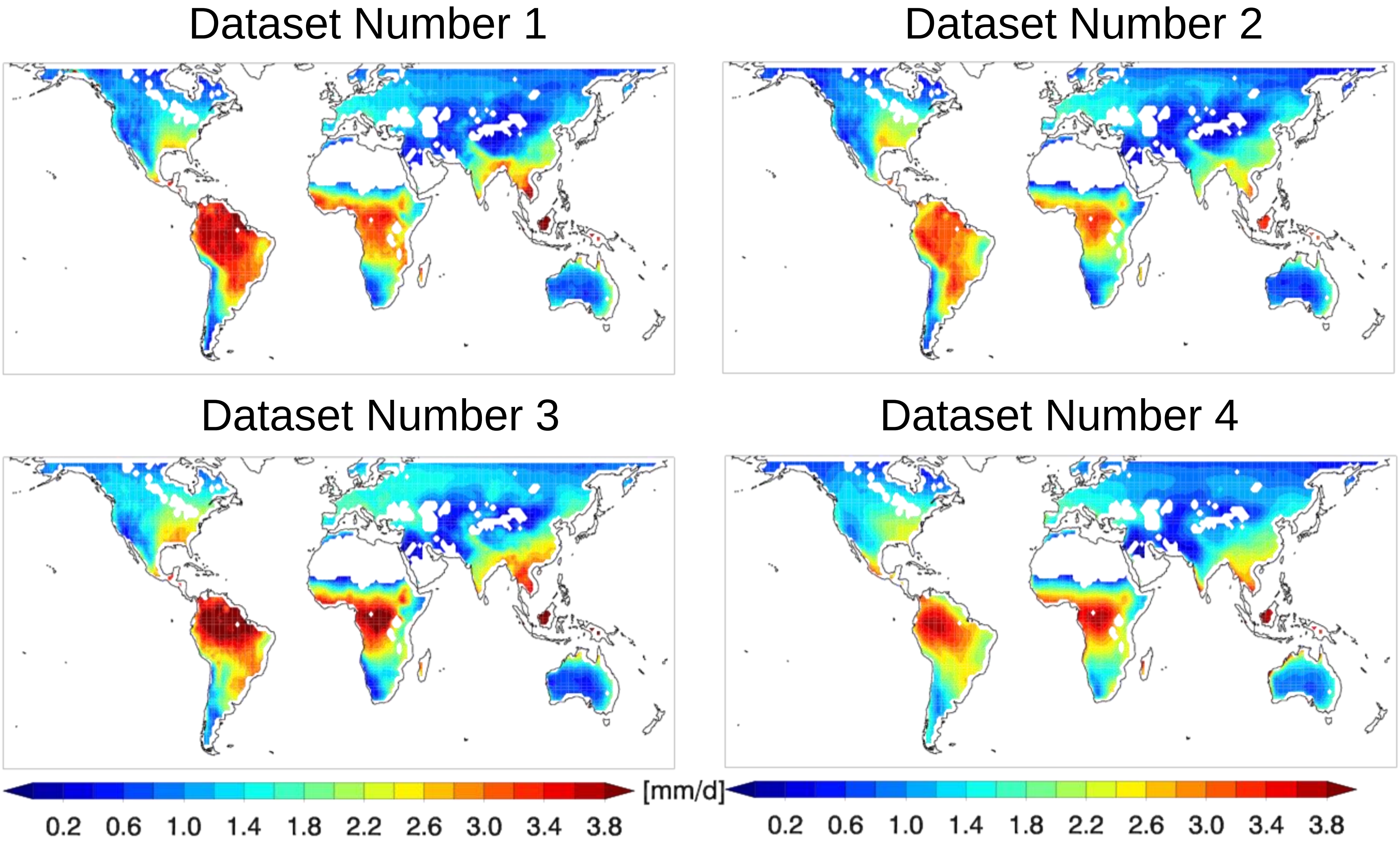


Data science with R

Brigitte Mueller

https://github.com/mbbrigitte/Ruby_Talk_Material

How to compare evaporation datasets?



Dataset Number 1

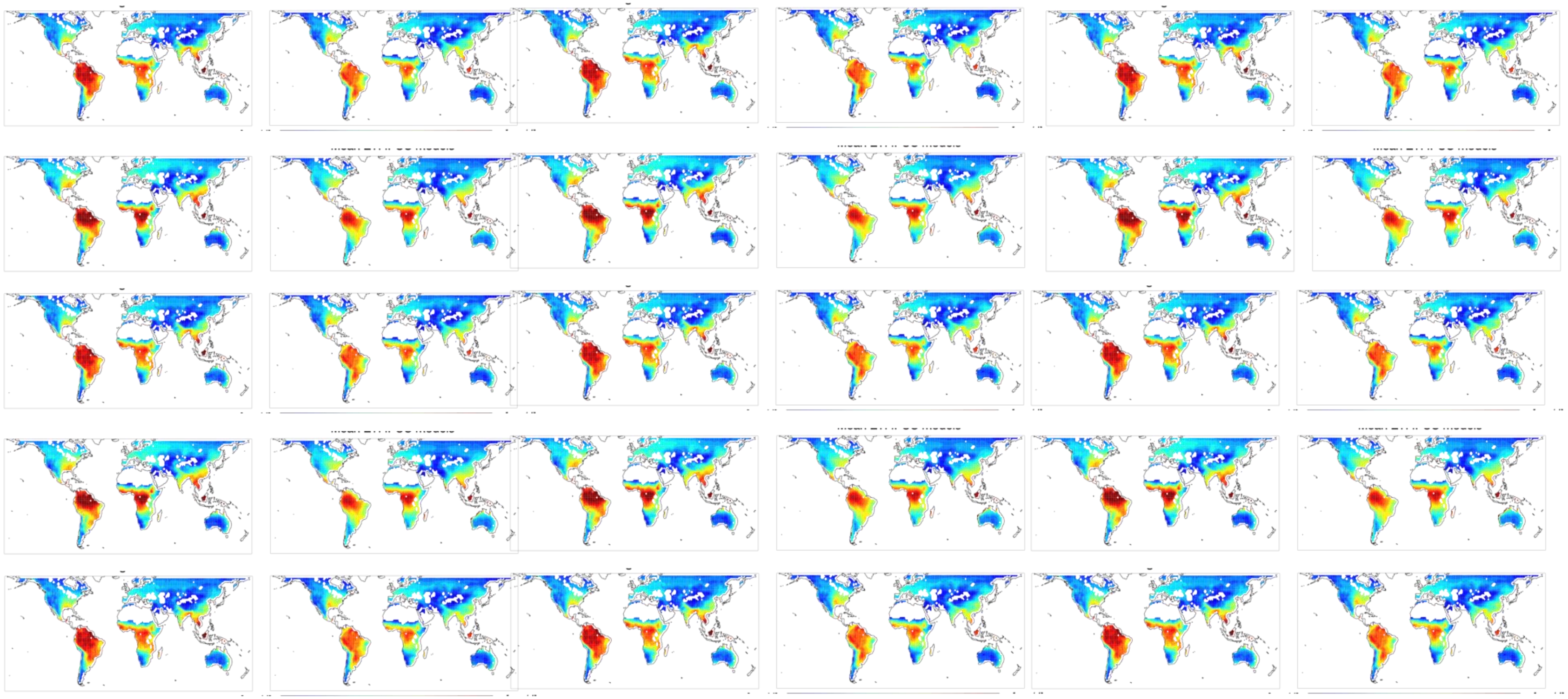
Dataset Number 2

Dataset Number 3

Dataset Number 4

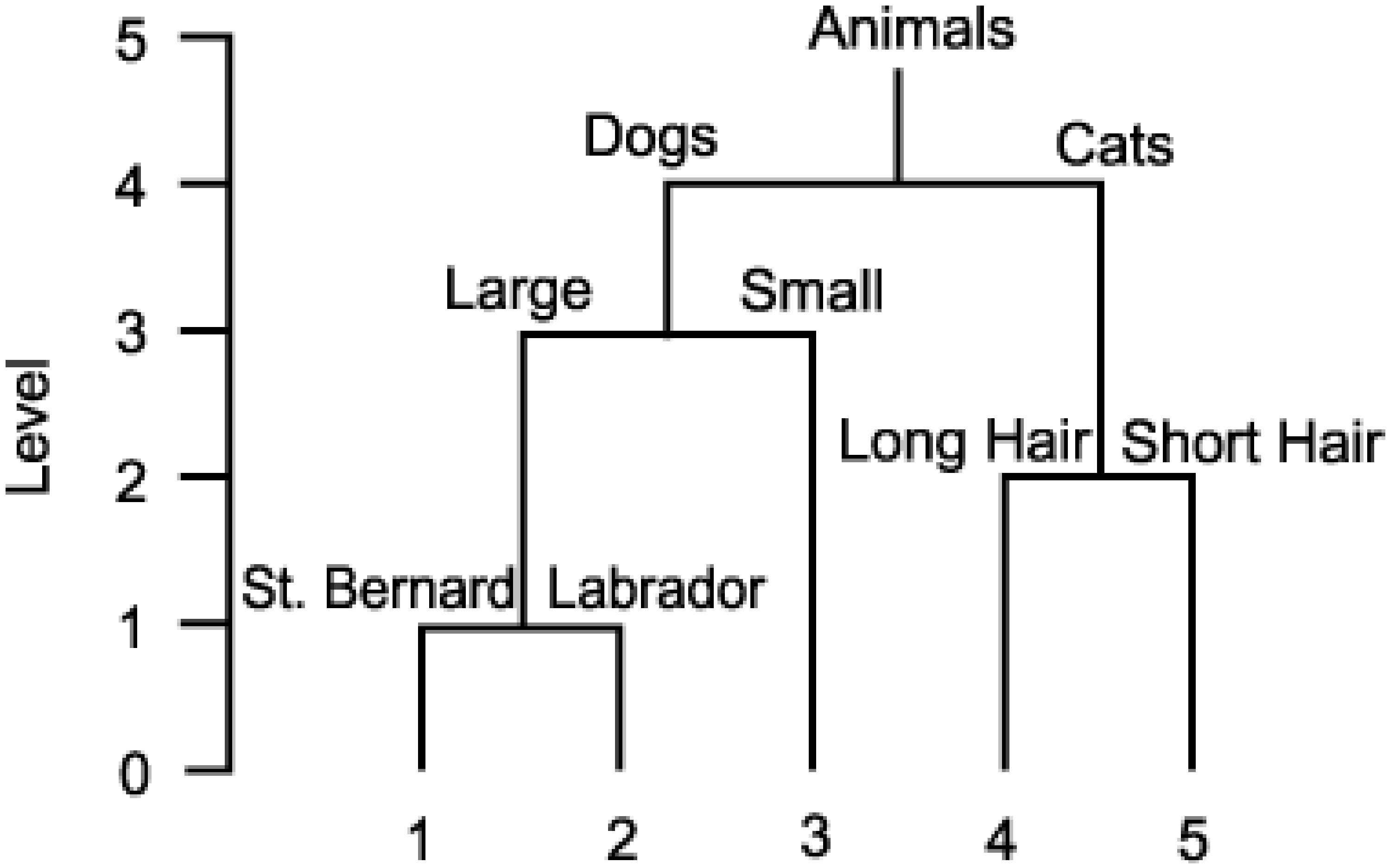
Mueller *et al*. *GRL*, 2011

How to compare 30 datasets?



→ Hierarchical clustering

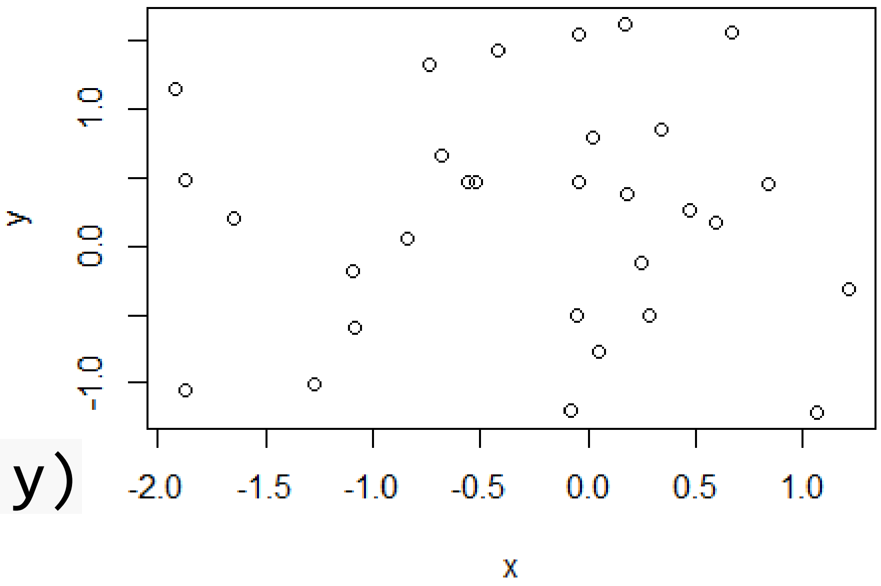
How to compare datasets?



Create some data

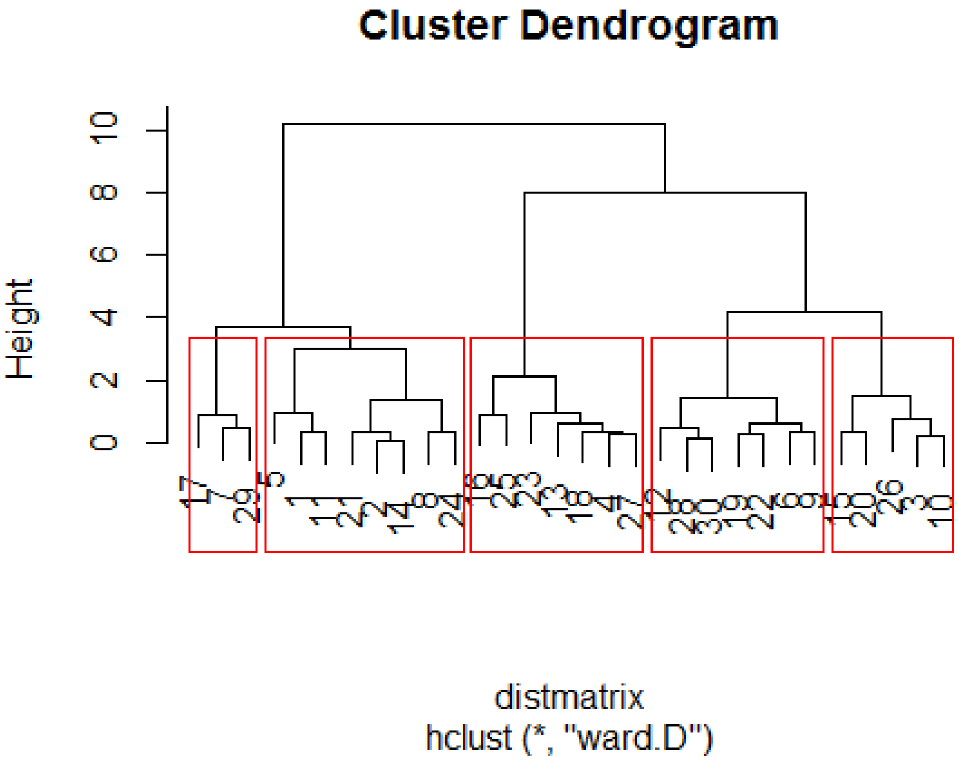
```
x <- rnorm(30)
y <- rnorm(30)
plot(x,y)

datamatrix <- cbind(x,y)
```

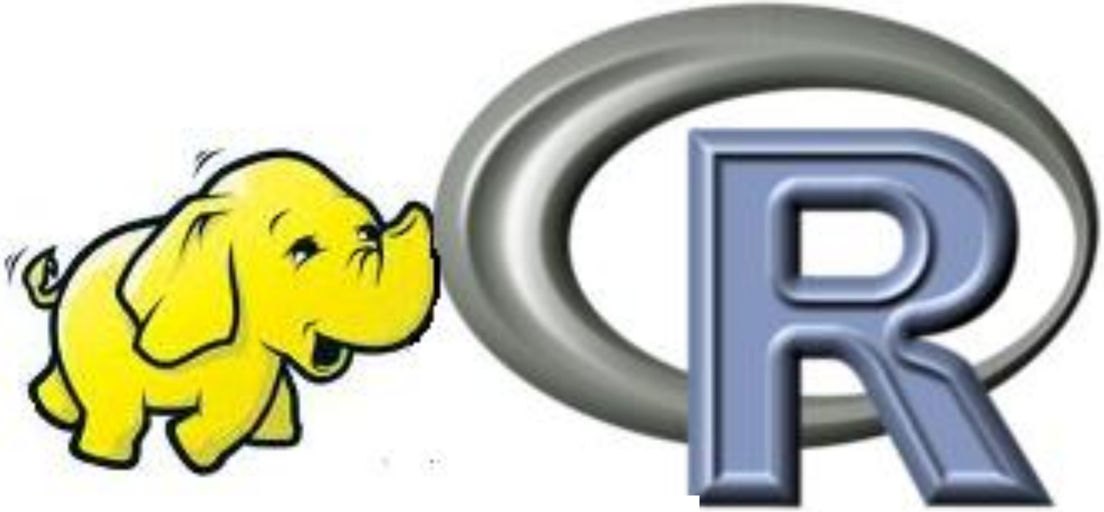


Calculate the distances and the clusters

```
distmatrix <- dist(datamatrix)
fit <- hclust(distmatrix, method="ward.D")
plot(fit)
```



R



Spark
R

R
in
Ruby

```
>> require "rinruby"
```

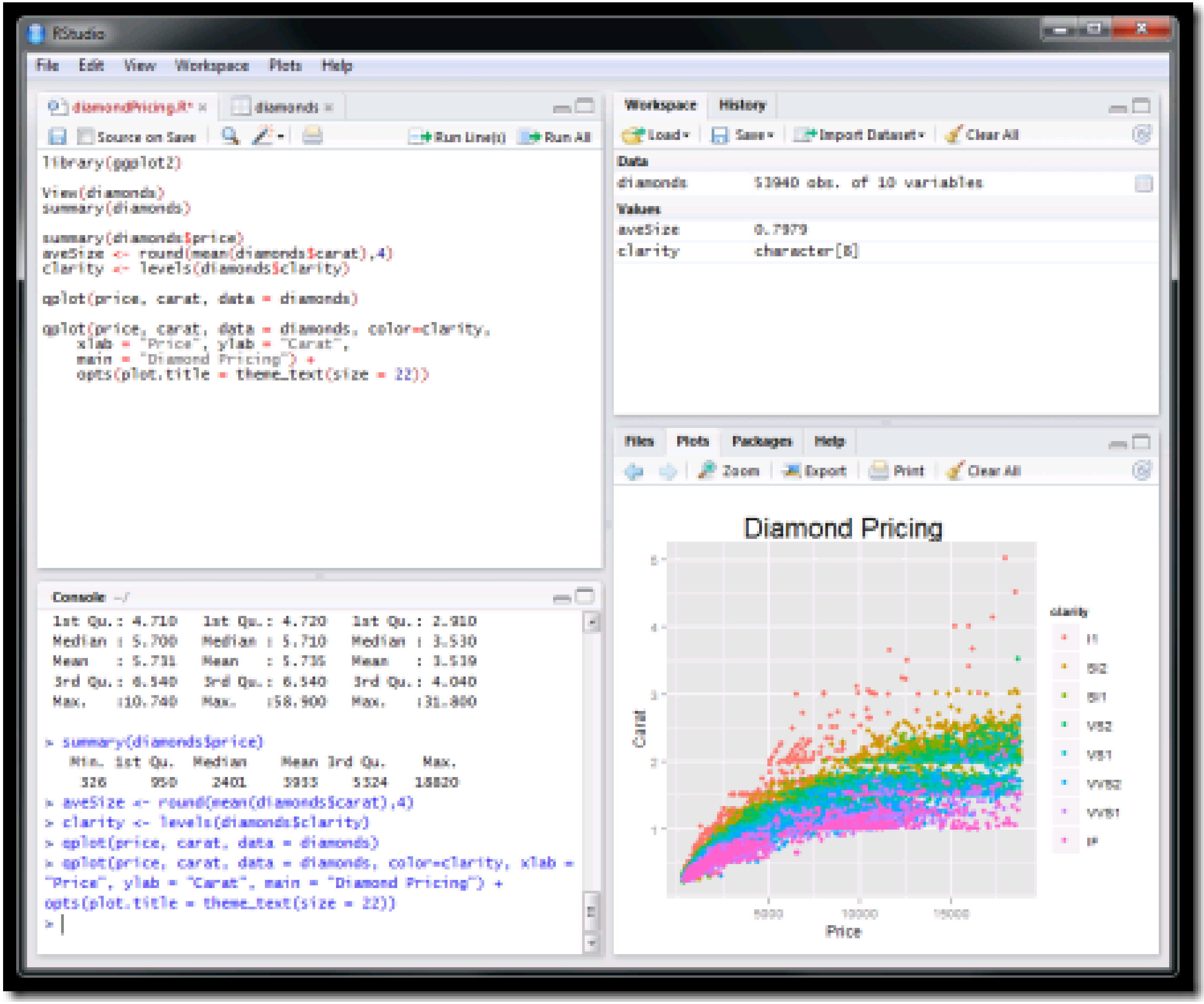
- Reads definition of RinRuby class into Ruby interpreter
- Creates instance of RinRuby class named R
- eval instance method passes R commands contained in the supplied string

```
>>  sample_size = 10
>>  R.eval "x <- rnorm(#{sample_size})"
>>  R.eval "summary(x)"
```

produces the following :

```
    Min.   1st Qu. Median   Mean     3rd Qu.   Max.
-1.88900 -0.84930 -0.45220 -0.49290 -0.06069 0.78160
```

More info: https://sites.google.com/a/ddahl.org/rinruby-users/documentation



```
library(ggplot2)

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat),4)
clarity <- levels(diamonds$clarity)

qplot(price, carat, data = diamonds)

qplot(price, carat, data = diamonds, color=clarity,
    xlab = "Price", ylab = "Carat",
    main = "Diamond Pricing") +
    opts(plot.title = theme_text(size = 22))
```

Data

diamonds	53940 obs. of 10 variables

Values

aveSize	0.7979
clarity	character[8]

```
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800

> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat),4)
> clarity <- levels(diamonds$clarity)
> qplot(price, carat, data = diamonds)
> qplot(price, carat, data = diamonds, color=clarity, xlab =
"Price", ylab = "Carat", main = "Diamond Pricing") +
opts(plot.title = theme_text(size = 22))
> |
```

Supervised learning

Delayed or not?

Target

Binary prediction: Delayed 0/1

Arriving late (= 15 minutes)

50% accuracy

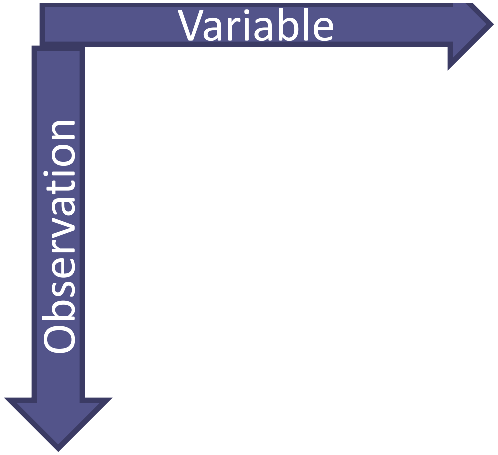
Goal
70% accuracy

Prepare data

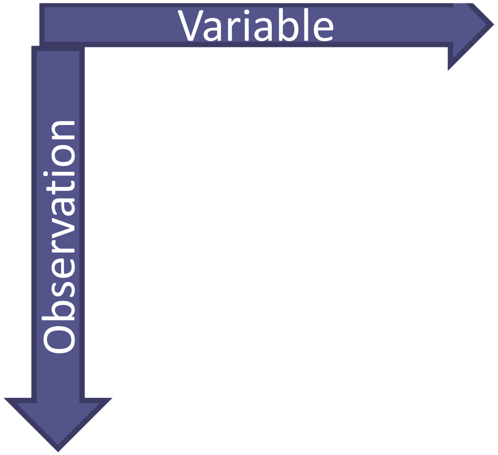
Clean, explore, tidy

Prepare data

Clean, explore, tidy

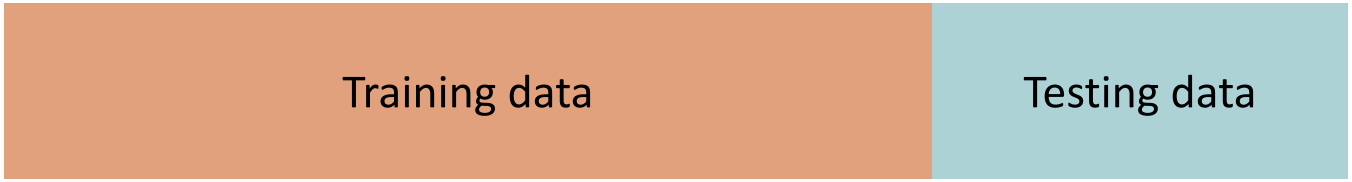


Prepare data

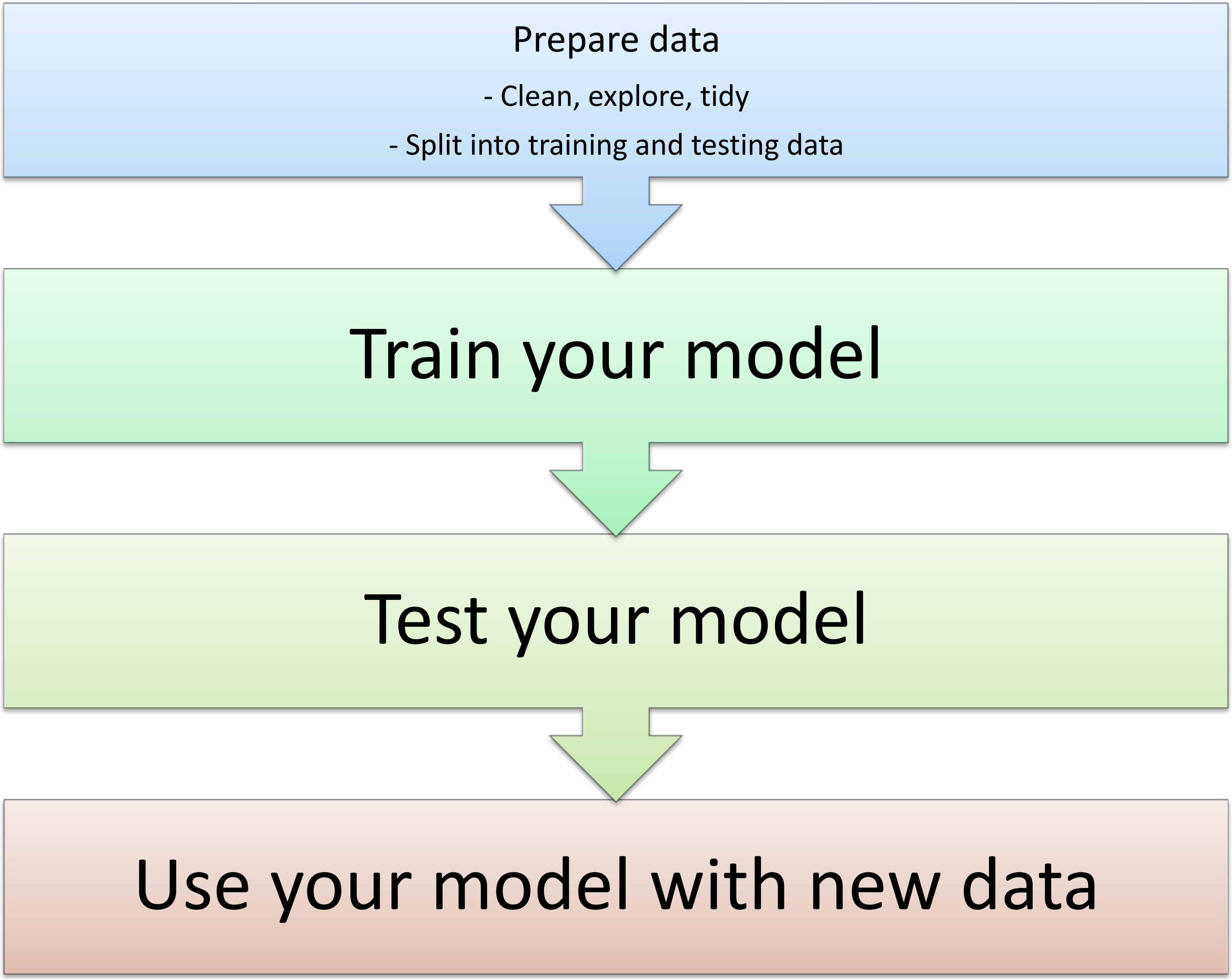


Clean, explore, tidy

Split into training and testing data


Training data

Testing data



Prepare data

- Clean, explore, tidy

- Split into training and testing data

Train your model

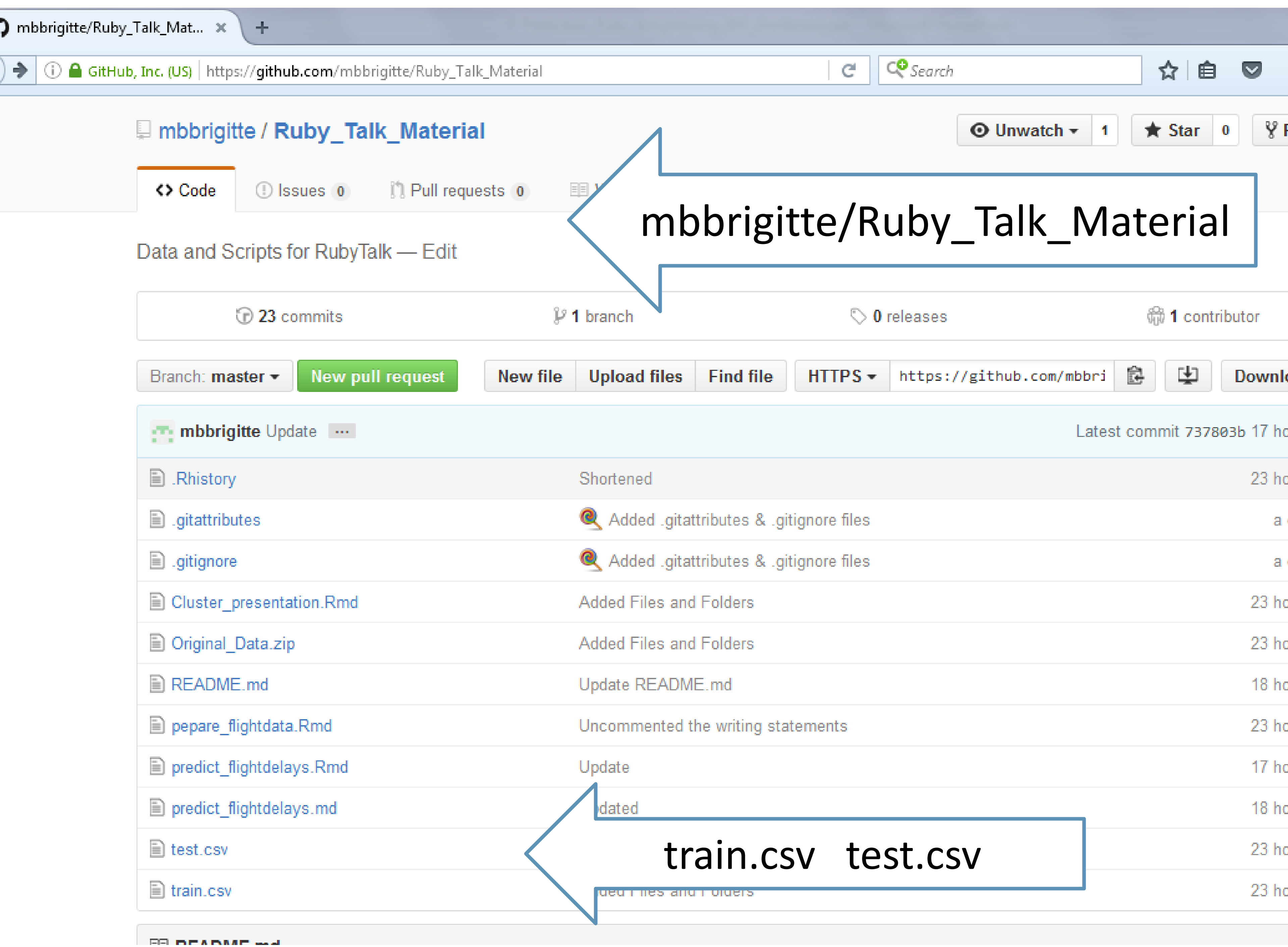
Test your model

Use your model with new data

Data

Prepared for you, tidied and saved as
train.csv test.csv

Download at
https://github.com/mbbrigitte/Ruby_Talk_Material



mbbrigitte / Ruby_Talk_Material

Data and Scripts for RubyTalk — Edit

23 commits | 1 branch | 0 releases | 1 contributor

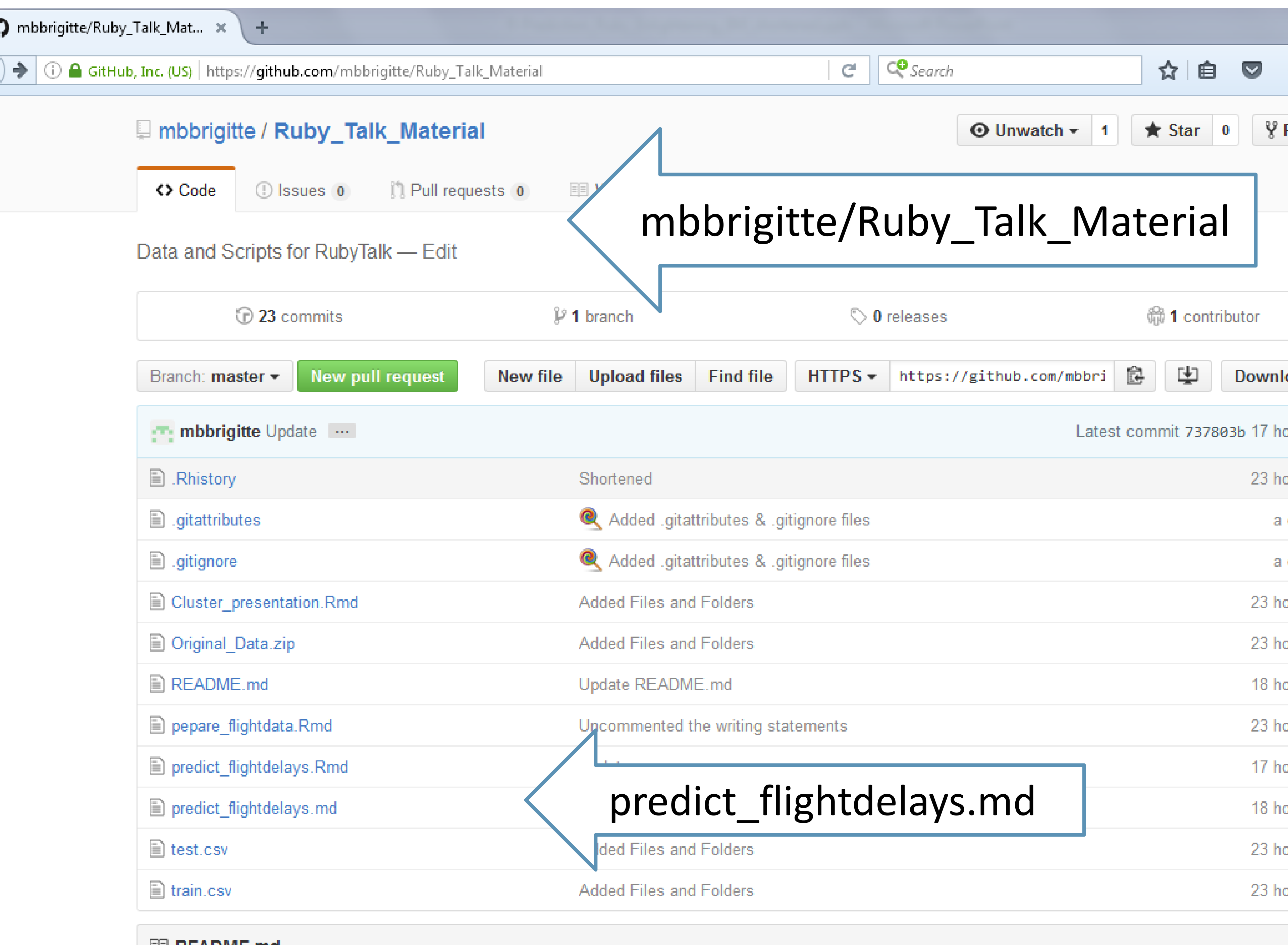
mbbrigitte Update ... Latest commit 737803b 17 ho

.Rhistory	Shortened	23 ho
.gitattributes	Added .gitattributes & .gitignore files	a
.gitignore	Added .gitattributes & .gitignore files	a
Cluster_presentation.Rmd	Added Files and Folders	23 ho
Original_Data.zip	Added Files and Folders	23 ho
README.md	Update README.md	18 ho
pepare_flightdata.Rmd	Uncommented the writing statements	23 ho
predict_flightdelays.Rmd	Update	17 ho
predict_flightdelays.md	dated	18 ho
test.csv		23 ho
train.csv	ded Files and Folders	23 ho

README.md

mbbrigitte/Ruby_Talk_Material

train.csv test.csv



mbbrigitte/Ruby_Talk_Material

predict_flightdelays.md

mbbrigitte/Ruby_Talk_Mat...

https://github.com/mbbrigitte/Ruby_Talk_Material

mbbrigitte / Ruby_Talk_Material

Data and Scripts for RubyTalk — Edit

23 commits · 1 branch · 0 releases · 1 contributor

Branch: master · https://github.com/mbbri

mbbrigitte Update ··· Latest commit 737803b 17 ho

.Rhistory	Shortened	23 ho
.gitattributes	Added .gitattributes & .gitignore files	a
.gitignore	Added .gitattributes & .gitignore files	a
Cluster_presentation.Rmd	Added Files and Folders	23 ho
Original_Data.zip	Added Files and Folders	23 ho
README.md	Update README.md	18 ho
pepare_flightdata.Rmd	Uncommented the writing statements	23 ho
predict_flightdelays.Rmd		17 ho
predict_flightdelays.md		18 ho
test.csv	...ded Files and Folders	23 ho
train.csv	Added Files and Folders	23 ho

README.md

Data

What variables are in the files?
Check with

```
read.csv(filename)
names(data)
```

ARR_DEL15, DAY_OF_WEEK, CARRIER, DEST, ORIGIN, DEP_TIME_BLK

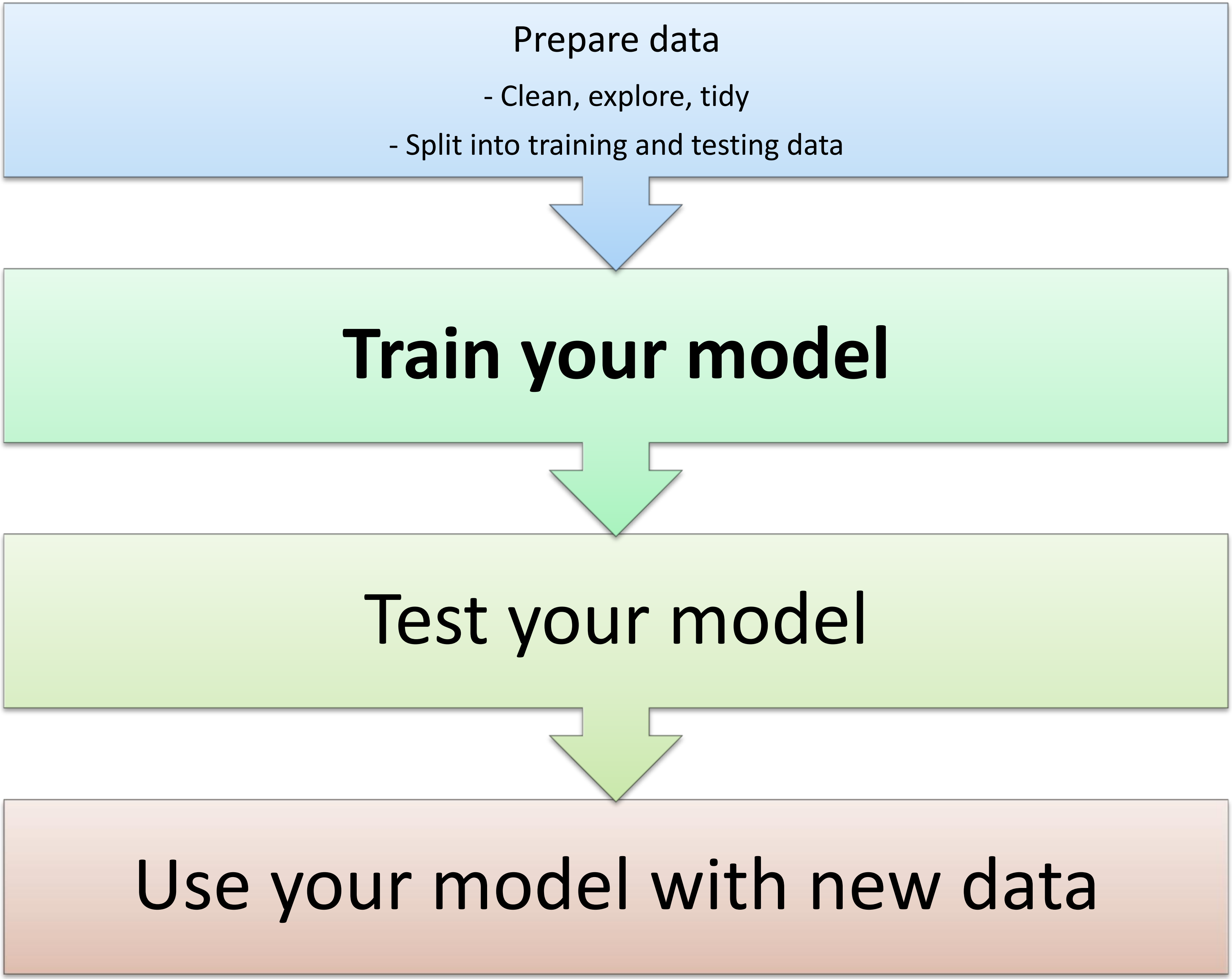
Code: Set-up

```
set.seed(100)

install.packages(‘caret’)
library(caret)
```

Code: Read data

```
trainData <- read.csv('train.csv',sep=',', header=TRUE)
testData <- read.csv('test.csv',sep=',', header=TRUE)
```



Prepare data

- Clean, explore, tidy

- Split into training and testing data

Train your model

Test your model

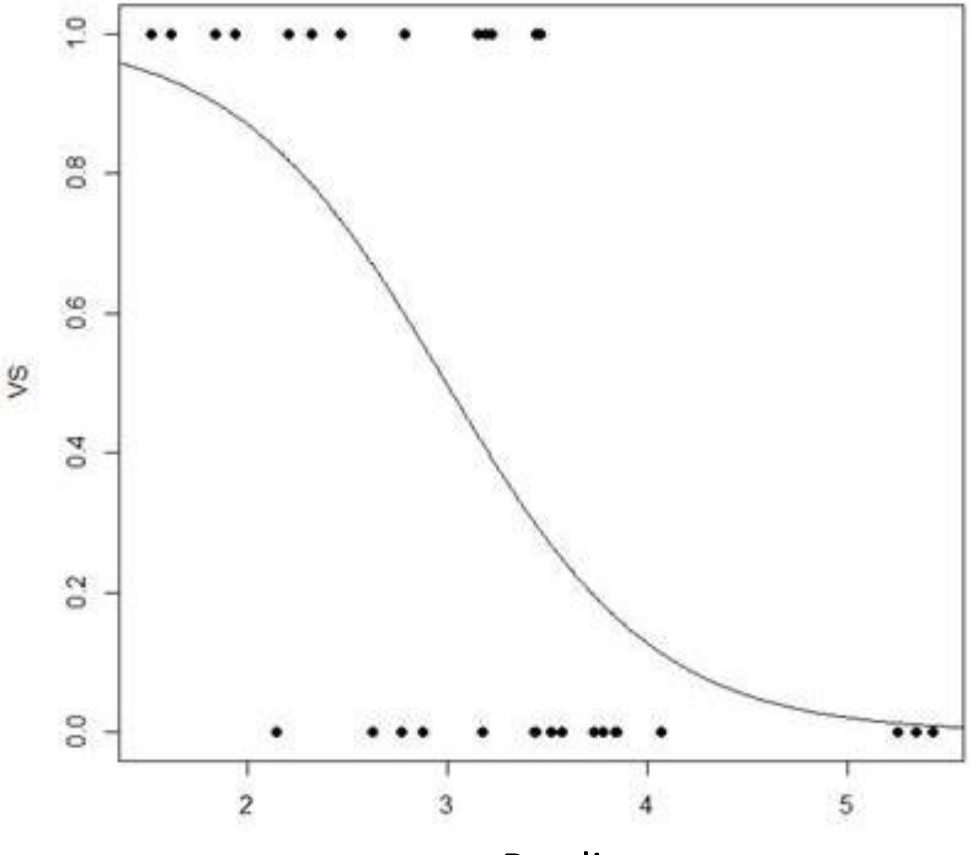
Use your model with new data

Select algorithm

- Classification algorithm
- Start simple
- If performance not that good, improve
  - Ensemble algorithms
  - Select more important variables from the data
  - Include additional predictor variables
  - Feature-engineering

Logistic regression

- Regression that predicts a categorical value



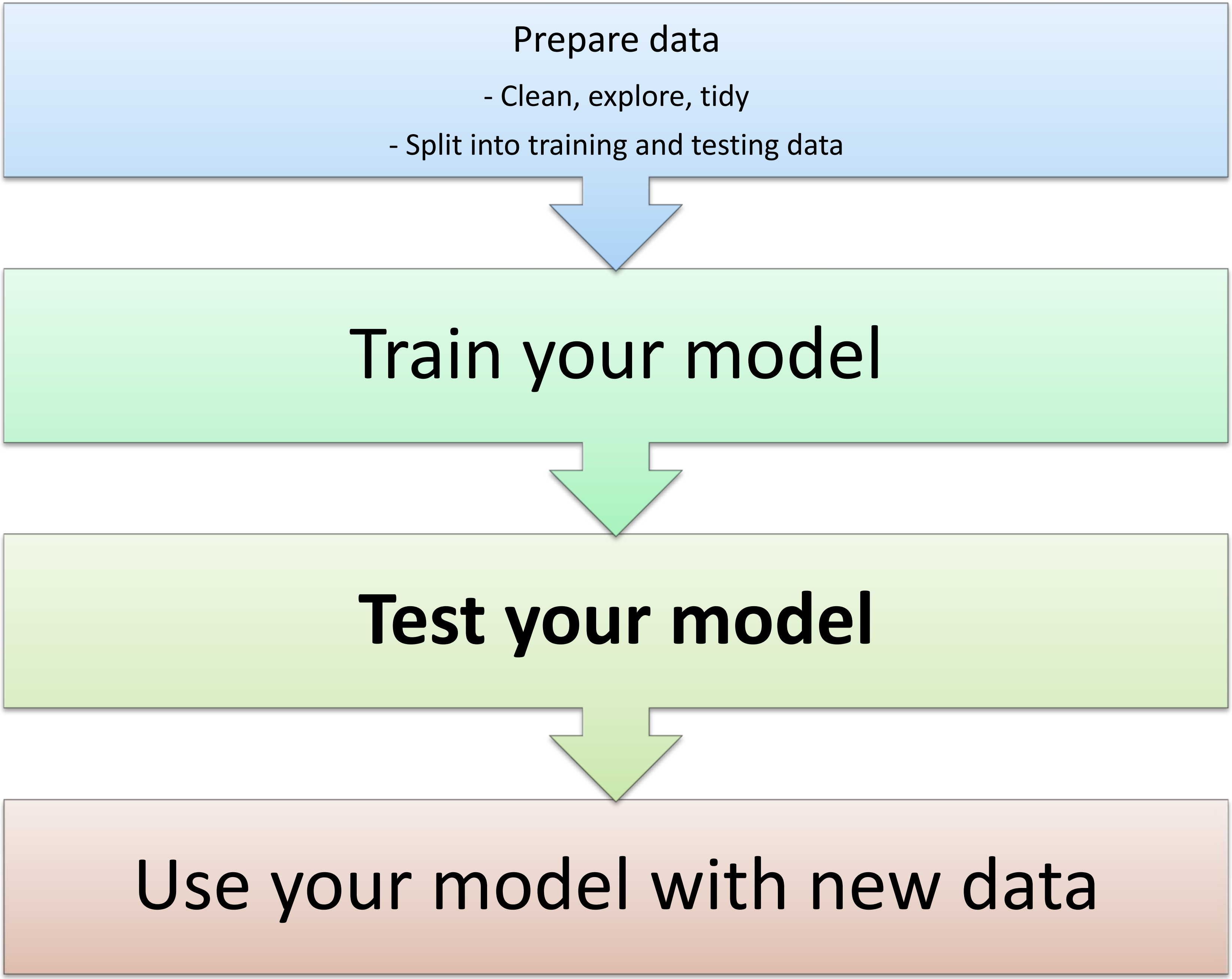
Predictor

Train

```
library(caret)
```

```
logisticRegModel <- train(ARR_DEL15 ~ .,
data=trainData, method = 'glm', family =
'binomial')
```

Dot: 'all available variables, i.e. all columns', glm generalized linear regression. Family binomial for logistic regression.



Prepare data

- Clean, explore, tidy
- Split into training and testing data

Train your model

Test your model

Use your model with new data

Predict and test

Use your model and the test data to check how well we predict flight arrival delays.

```
logRegPrediction <- predict(logisticRegModel, testData)

logRegConfMat <- confusionMatrix(logRegPrediction,
                testData[,"ARR_DEL15"])

logRegConfMat
```

```
## Confusion Matrix and Statistics
##                 Reference
## Prediction    0    1
##          0    7465 2273
##          1      65   94
##
##                Accuracy : 0.7638
##                  95% CI : (0.7553, 0.7721)
##     No Information Rate : 0.7608
##     P-Value [Acc > NIR] : 0.2513
##
##                   Kappa : 0.0457
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99137
##             Specificity : 0.03971
##          Pos Pred Value : 0.76658
##          Neg Pred Value : 0.59119
##              Prevalence : 0.76084
##          Detection Rate : 0.75427
##    Detection Prevalence : 0.98393
##       Balanced Accuracy : 0.51554
##
##        'Positive' Class : 0
## Confusion Matrix and Statistics
##                 Reference
## Prediction    0    1
##          0 7465 2273
##          1   65   94
##
##                Accuracy : 0.7638
##                  95% CI : (0.7553, 0.7721)
##     No Information Rate : 0.7608
##     P-Value [Acc > NIR] : 0.2513
##
##                   Kappa : 0.0457
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99137
##             Specificity : 0.03971
##          Pos Pred Value : 0.76658
##          Neg Pred Value : 0.59119
##              Prevalence : 0.76084
##          Detection Rate : 0.75427
##    Detection Prevalence : 0.98393
##       Balanced Accuracy : 0.51554
##
##        'Positive' Class : 0
## Confusion Matrix and Statistics
##                Reference
## Prediction    0    1
##          0       7465 2273
##          1          65   94
##
##              Accuracy : 0.7638
##                95% CI : (0.7553, 0.7
##    No Information Rate : 0.7608
##    P-Value [Acc > NIR] : 0.2513
##
##                 Kappa : 0.0457
##  Mcnemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.99137
##           Specificity : 0.03971
##        Pos Pred Value : 0.76658
##        Neg Pred Value : 0.59119
##            Prevalence : 0.76084
##        Detection Rate : 0.75427
##   Detection Prevalence : 0.98393
##      Balanced Accuracy : 0.51554
##
##       'Positive' Class : 0
```

		Reference	
		0 not delayed	1 delayed
Prediction	0 not delayed	7465	2273
	1 delayed	64	94

Specificity = 94/(2273+94)

Specificity
proportion of negatives that are correctly identified as such

Specificity is low - Improve model

```
names(getModelInfo())

logisticRegModel <- train(ARR_DEL15 ~ .,
data=trainData, method = 'glm', family =
'binomial')
```

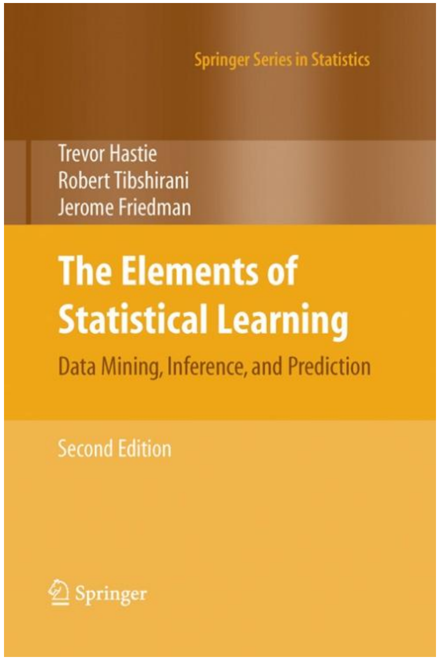
HB-JMN
SWISS

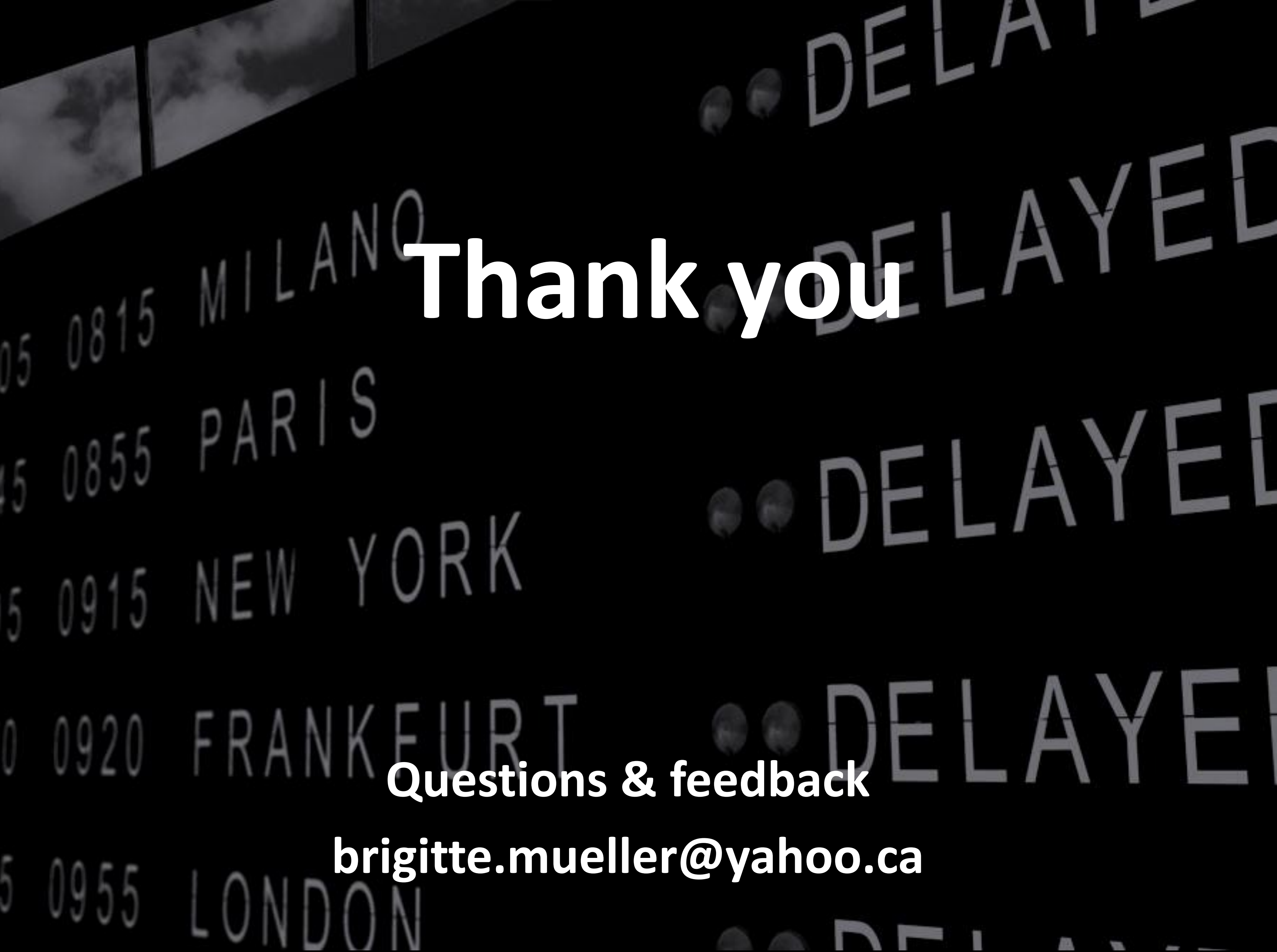
Next steps

- Try basics yourself
  - Improve model used with data in this talk
  - Titanic dataset: http://amunategui.github.io/binary-outcome-modeling/
  - https://www.datacamp.com/courses/kaggle-tutorial-on-machine-learing-the-sinking-of-the-titanic
- Try advanced methods
  - Kaggle
- Find your own dataset
- Learn more about machine learning and R:

Further reading

Elements of Statistical Learning, Hastie et al. 2009, Springer:
Available for fee
http://statweb.stanford.edu/~tibs/ElemStatLearn/


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer



Thank you

Questions & feedback

brigitte.mueller@yahoo.ca

Picture sources

- http://www.tronviggroup.com/open-source-evolution/ (world with people)
- http://twit88.com/blog/2011/03/01/open-source-ide-for-r/ (R IDE)
- http://www.dailymail.co.uk (Coin toss)
- http://www.theanalysisfactor.com/r-glm-plotting/ (log. Regression figure)

Do it yourself

- Download and install R https://www.r-project.org/ and RStudio https://www.rstudio.com/ if you want to (it is convenient)
- Download the train.csv and test.csv files from Github https://github.com/mbbrigitte/Ruby_Talk_Material
- Use the ….Rmd files in R or just browse the code with the ….md file in your explorer

R: Packages and functions

- Lots of statistical packages (libraries)

```
install.packages(‘caret’)

library(caret)
```

- Run line by line or write programms with ending .R

```
source(“foo.R”)
```

- Function

```
myfun<- function(arg1, arg2, …)
     w=arg1^2
     return(arg2 + w)
     }
  myfun(arg=3,arg2=5)
```

R: Subsetting

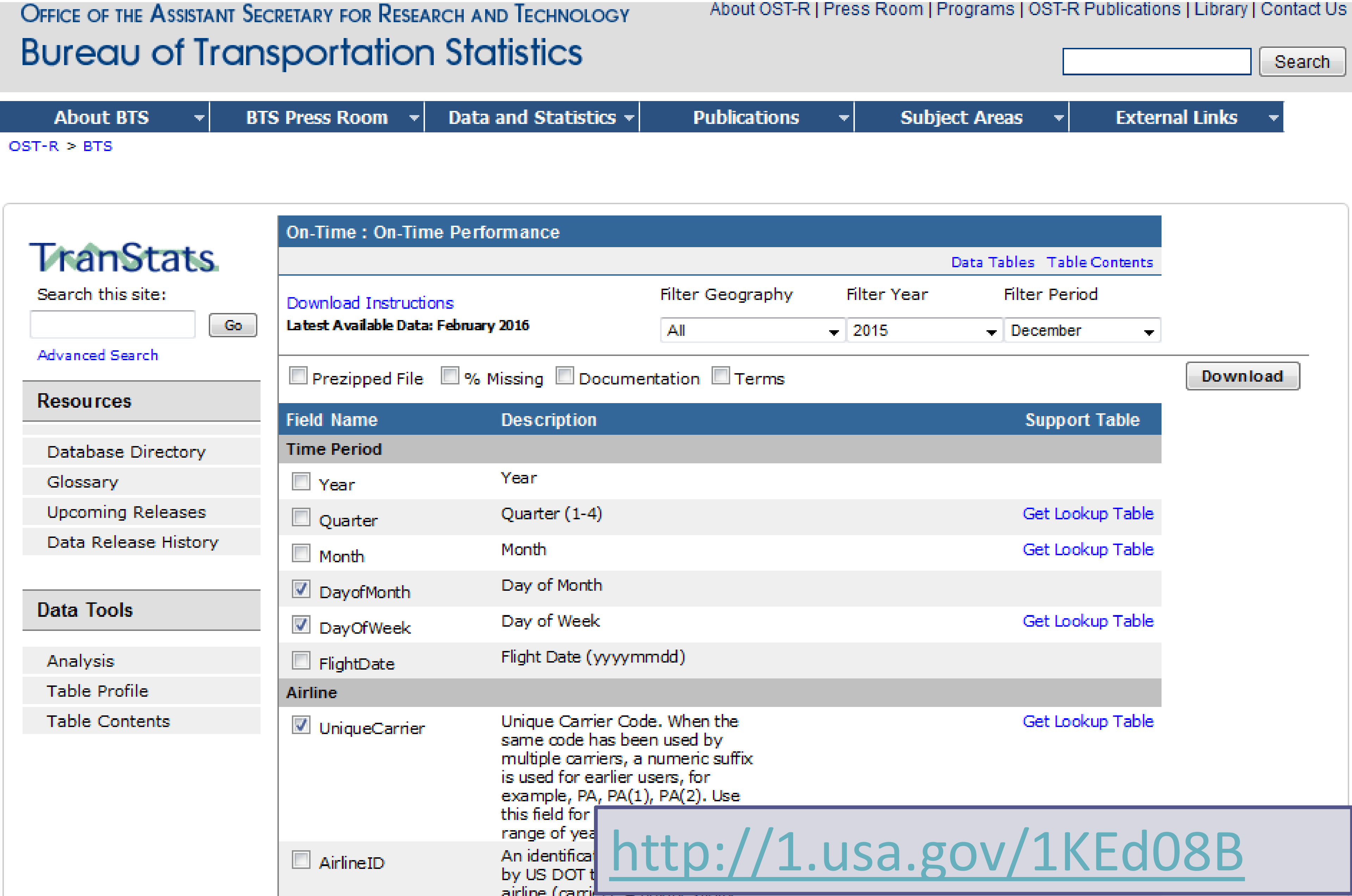
- Matrix

```
mat <- matrix(data=c(9,2,3,4,5,6),ncol=3)
mat[1,2] #output is 3
mat[2,] #output is 2,4,6
```

- Lists:

```
L = list(one=1, two=c(1,2), five=seq(0, 1,length=5))
L$five  #output 0.00 0.25 0.50 0.75 1.00
```

Original data source



On-Time : On-Time Performance

Latest Available Data: February 2016

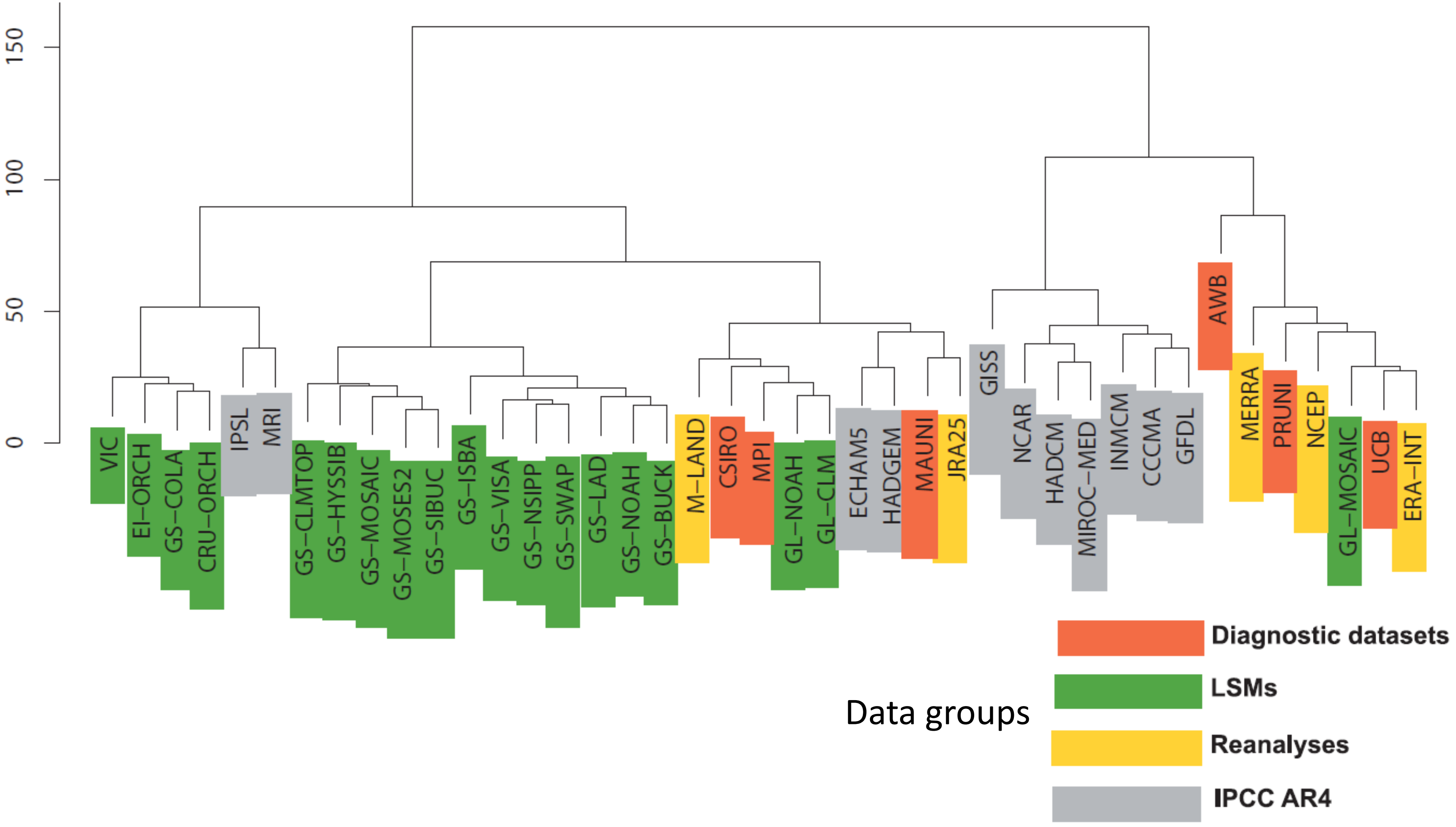
Filter Geography: All
Filter Year: 2015
Filter Period: December

□ Prezipped File □ % Missing □ Documentation □ Terms

Field Name	Description	Support Table
Time Period		
□ Year	Year	
□ Quarter	Quarter (1-4)	Get Lookup Table
□ Month	Month	Get Lookup Table
☑ DayofMonth	Day of Month	
☑ DayOfWeek	Day of Week	Get Lookup Table
□ FlightDate	Flight Date (yyyymmdd)	
Airline		
☑ UniqueCarrier	Unique Carrier Code. When the same code has been used by multiple carriers, a numeric suffix is used for earlier users, for example, PA, PA(1), PA(2). Use this field for range of yea	Get Lookup Table
□ AirlineID	An identifica by US DOT t airline (carri	

http://1.usa.gov/1KEd08B

Results from evaporation dataset clustering



Mueller *et al*. *GRL*, 2011

Example with gbm instead of glm method, i.e. boosted tree model: see http://topepo.github.io/caret/training.html

```
fitControl <- trainControl(method = 'repeatedcv', number = 10, repeats = 10)


gbmFit1 <- train(ARR_DEL15 ~ ., data=trainData, method = 'gbm',trControl = fitControl,verbose = FALSE)
```