R by example: mining Twitter for consumer attitudes towards airlines

presented at the

Boston Predictive Analytics MeetUp

by

Jeffrey Breen

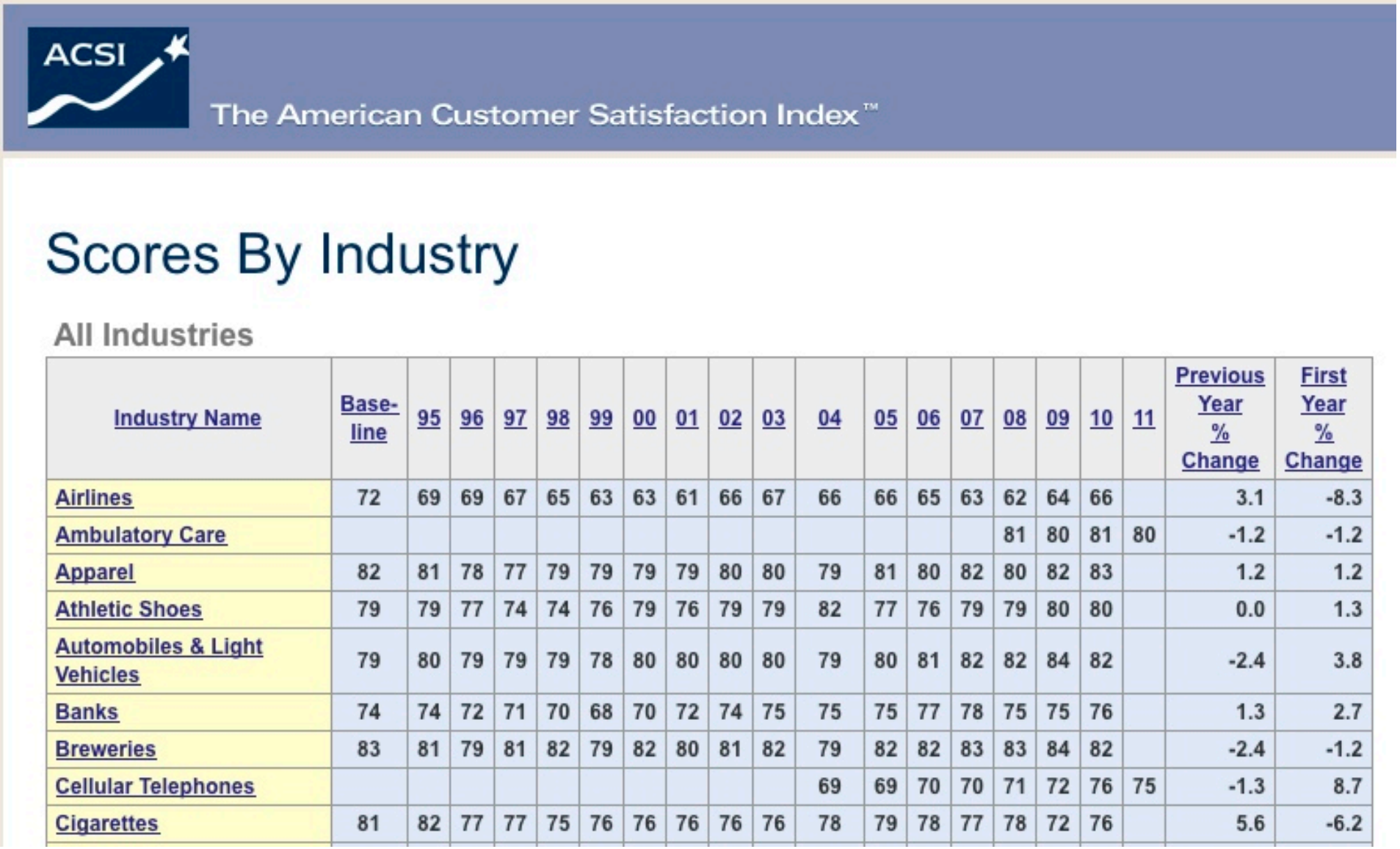
President
Cambridge Aviation Research

jbreen@cambridge.aero

June 2011

Cambridge Aviation Research • 245 First Street • Suite 1800 • Cambridge, MA 02142 • cambridge.aero

Airlines top customer satisfaction… alphabetically



Scores By Industry

All Industries

Industry Name	Base-line	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	Previous Year % Change	First Year % Change
Airlines	72	69	69	67	65	63	63	61	66	67	66	66	65	63	62	64	66		3.1	-8.3
Ambulatory Care															81	80	81	80	-1.2	-1.2
Apparel	82	81	78	77	79	79	79	79	80	80	79	81	80	82	80	82	83		1.2	1.2
Athletic Shoes	79	79	77	74	74	76	79	76	79	79	82	77	76	79	79	80	80		0.0	1.3
Automobiles & Light Vehicles	79	80	79	79	79	78	80	80	80	80	79	80	81	82	82	84	82		-2.4	3.8
Banks	74	74	72	71	70	68	70	72	74	75	75	75	77	78	75	75	76		1.3	2.7
Breweries	83	81	79	81	82	79	82	80	81	82	79	82	82	83	83	84	82		-2.4	-1.2
Cellular Telephones											69	69	70	70	71	72	76	75	-1.3	8.7
Cigarettes	81	82	77	77	75	76	76	76	76	76	78	79	78	77	78	72	76		5.6	-6.2

http://www.theacsi.org/

Actually, they rank below the Post Office and health insurers

Hotels	75	73	72	71	71	72	72	71	71	73	72	73	75	71	75	75	75		0.0	0.0
Limited Service Restaurants	69	70	66	68	69	69	70	71	71	74	NM**	76	77	77	78	78	75		-3.8	8.7
Supermarkets	76	75	74	73	73	74	73	75	75	74	73	74	75	76	76	76	75		-1.3	-1.3
Fixed Line Telephone Service	81	80	79	75	74	73	72	70	71	72	71	70	70	70	73	72	75	73	-2.7	-9.9
Internet News & Information									73	74	75	75	73	75	75	74	74		0.0	1.4
Network Cable TV News	77	76	70	62	65	62	64	62	65	68	66	68	69	67	69	71	74	77	4.1	0.0
Health Insurance								68	69	70	67	68	72	71	73	75	73		-2.7	7.4
Hospitals	74	74	71	67	72	70	69	68	70	73	76	71	74	77	75	77	73	77	5.5	4.1
Wireless Telephone Service											65	63	66	68	68	69	72	71	-1.4	9.2
U.S. Postal Service	61	69	74	69	71	71	72	70	73	72	74	73	71	73	74	74	71		-4.1	16.4
Internet Social Media																	70		N/A	N/A
Gasoline Stations	78	80	77	78	79	76	75	77	76	75	70	69	71	70	74	76	70		-7.9	-10.3
Subscription Television Service								64	61	61	61	61	63	62	64	63	66	66	0.0	3.1
Airlines	72	69	69	67	65	63	63	61	66	67	66	66	65	63	62	64	66		3.1	-8.3
Newspapers	72	68	69	69	66	69	68	68	63	64	68	63	63	66	64	63	65	65	0.0	-9.7
Municipal Utilities																		73	N/A	N/A
Investor-Owned Utilities																		74	N/A	N/A
Cooperative Utilities																		82	N/A	N/A

which gives us plenty to listen to

RT @dave_mcgregor: Publicly pledging to never fly @delta again. The worst airline ever. U have lost my patronage forever due to ur incompetence

Completely unimpressed with @continental or @united. Poor communication, goofy reservations systems and all to turn my trip into a mess.

@united #fail on wifi in red carpet clubs (too slow), delayed flight, customer service in red carpet club (too slow), hmmm do u see a trend?

@United Weather delays may not be your fault, but you are in the customer service business. It's atrocious how people are getting treated!

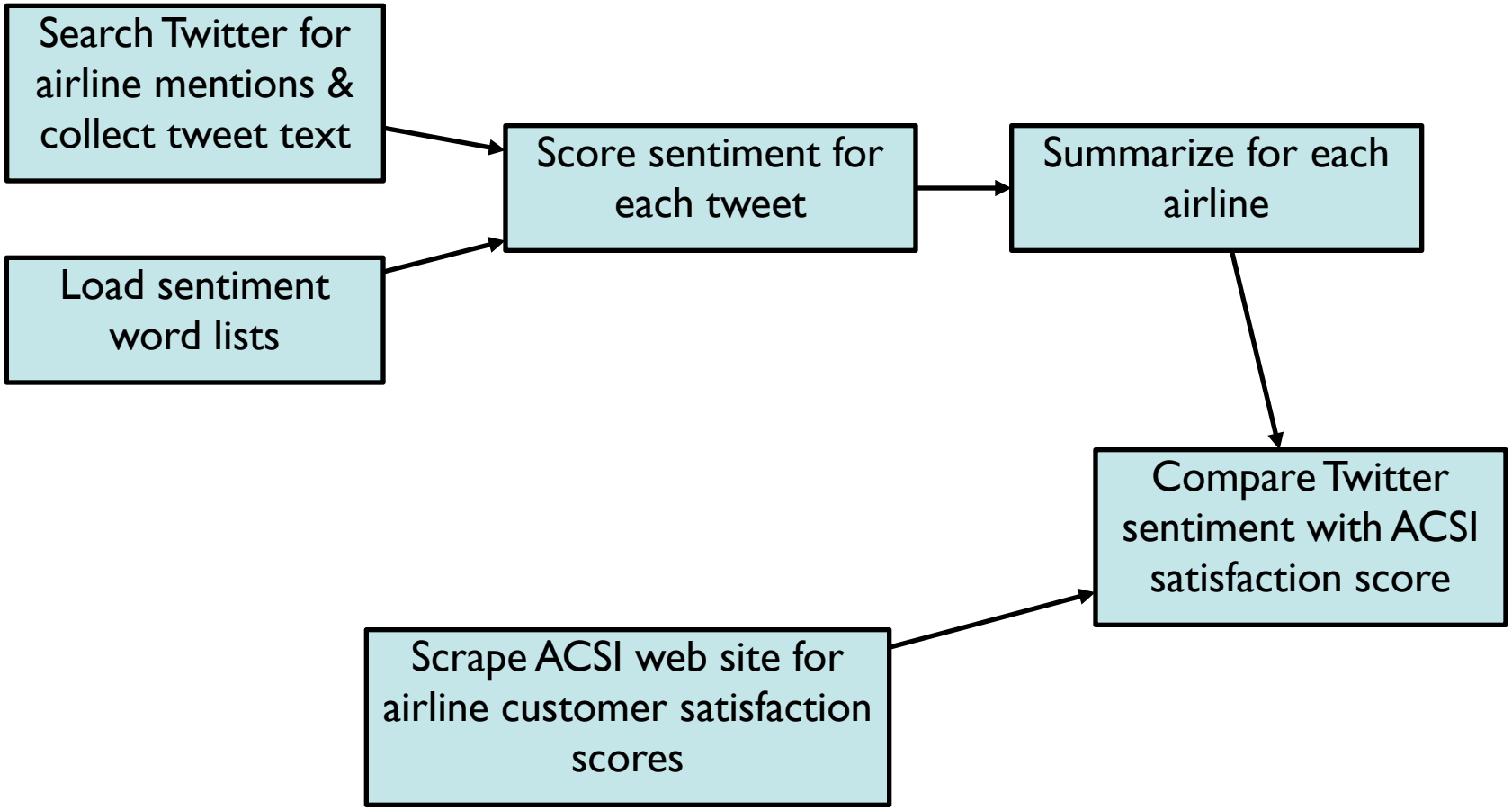
We were just told we are delayed 1.5 hrs & next announcement on @JetBlue - "We're selling headsets." Way to capitalize on our misfortune.

@SouthwestAir I know you don't make the weather. But at least pretend I am not a bother when I ask if the delay will make miss my connection

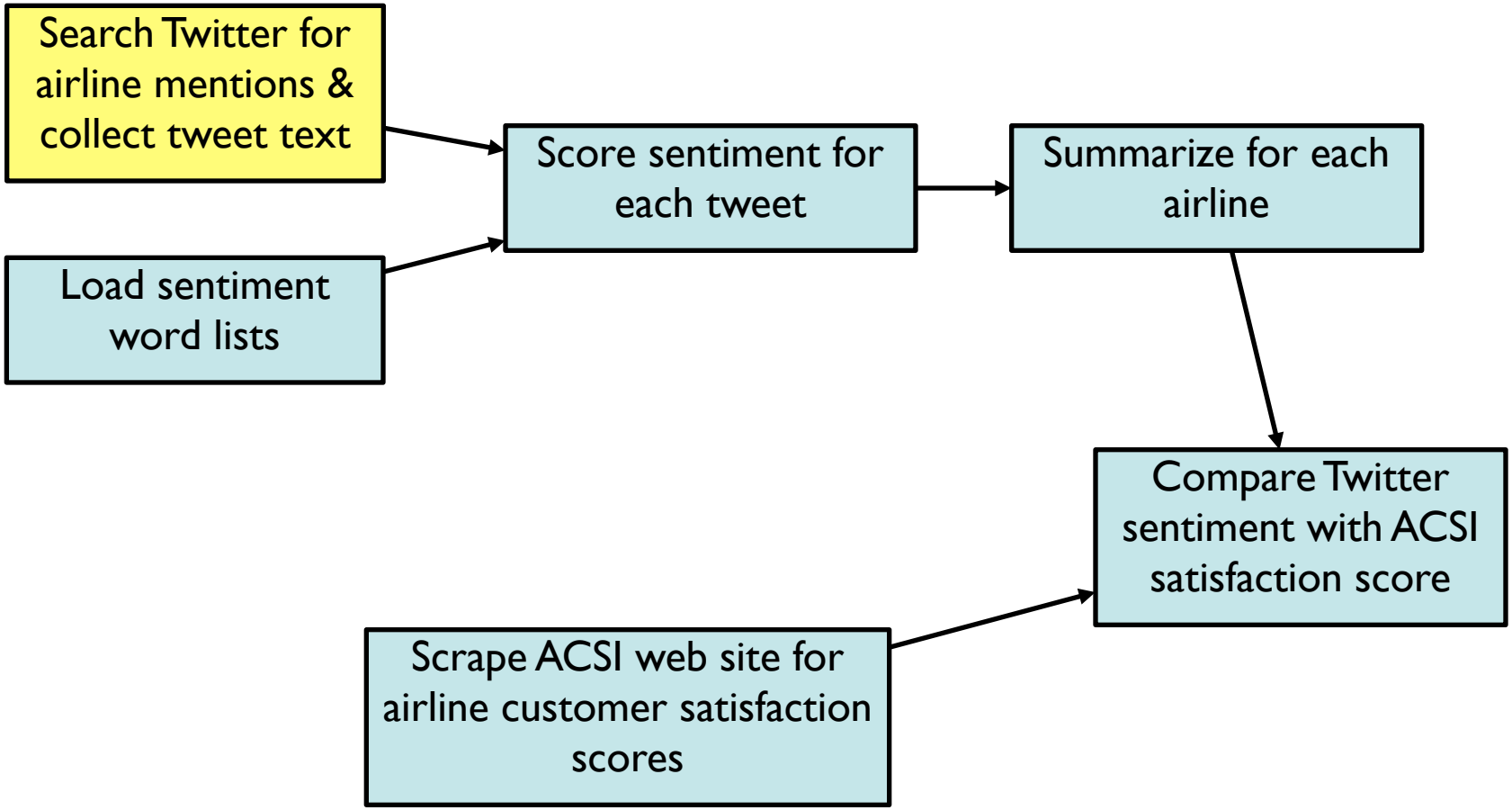
@SouthwestAir
I hate you with every single bone in my body for delaying my flight by 3 hours, 30mins before I was supposed to board. #hate

Hey @delta - you suck! Your prices are over the moon & to move a flight a cpl of days is $150.00. Insane. I hate you! U ruined my vacation!

Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Searching Twitter in one line

R's XML and RCurl packages make it easy to grab web data, but Jeff Gentry's twitteR package makes searching Twitter almost too easy:

```
> # load the package
> library(twitteR)
> # get the 1,500 most recent tweets mentioning '@delta':
> delta.tweets = searchTwitter('@delta', n=1500)
```

See what we got in return:

```
> length(delta.tweets)
[1] 1500
> class(delta.tweets)
[1] "list"
```

A "list" in R is a collection of objects and its elements may be named or just numbered.

"[[ ]]" is used to access elements.

Examine the output

Let's take a look at the first tweet in the output list:

```
> tweet = delta.tweets[[1]]
> class(tweet)
[1] "status"
attr(,"package")
[1] "twitteR"
```

tweet is an object of type "status" from the "twitteR" package.

It holds all the information about the tweet returned from Twitter.

The help page ("`?status`") describes some accessor methods like `getScreenName()` and `getText()` which do what you would expect:

```
> tweet$getScreenName()
[1] "Alaqawari"
> tweet$getText()
[1] "I am ready to head home. Inshallah will try to get on the earlier
flight to Fresno. @Delta @DeltaAssist"
```

Extract the tweet text

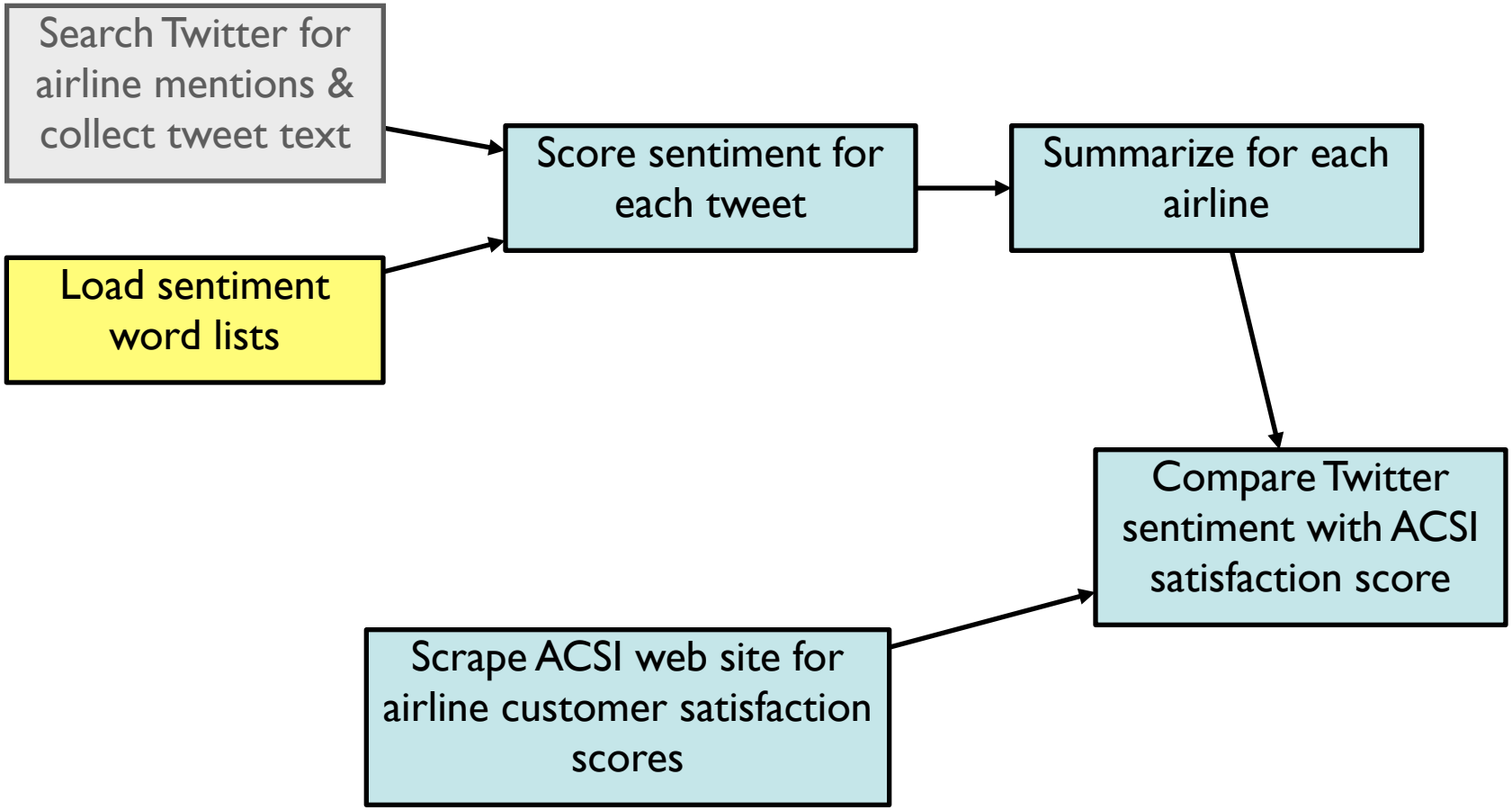
R has several (read: too many) ways to apply functions iteratively.

- The plyr package unifies them all with a consistent naming convention.
- The function name is determined by the input and output data types. We have a list and would like a simple array output, so we use “laply”:

```
> delta.text = laply(delta.tweets, function(t) t$getText() )
```

```
> length(delta.text)[1] 1500
> head(delta.text, 5)
[1] "I am ready to head home. Inshallah will try to get on the earlier
flight to Fresno. @Delta @DeltaAssist"
[2] "@Delta Releases 2010 Corporate Responsibility Report - @PRNewswire
(press release) : http://tinyurl.com/64mz3oh"
[3] "Another week, another upgrade! Thanks @Delta!"
[4] "I'm not able to check in or select a seat for flight DL223/KL6023 to
Seattle tomorrow. Help? @KLM @delta"
[5] "In my boredom of waiting realized @deltaairlines is now @delta
seriously..... Stil waiting and your not even unloading status yet"
```

Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Estimating Sentiment

There are many good papers and resources describing methods to estimate sentiment. These are very complex algorithms.

For this tutorial, we use a very simple algorithm which assigns a score by simply counting the number of occurrences of "positive" and "negative" words in a tweet. The code for our `score.sentiment()` function can be found at the end of this deck.

Hu & Liu have published an "opinion lexicon" which categorizes approximately 6,800 words as positive or negative and which can be downloaded.

Positive: love, best, cool, great, good, amazing

Negative: hate, worst, sucks, awful, nightmare

Load sentiment word lists

1. Download Hu & Liu's opinion lexicon:

http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

2. Loading data is one of R's strengths. These are simple text files, though they use ";" as a comment character at the beginning:

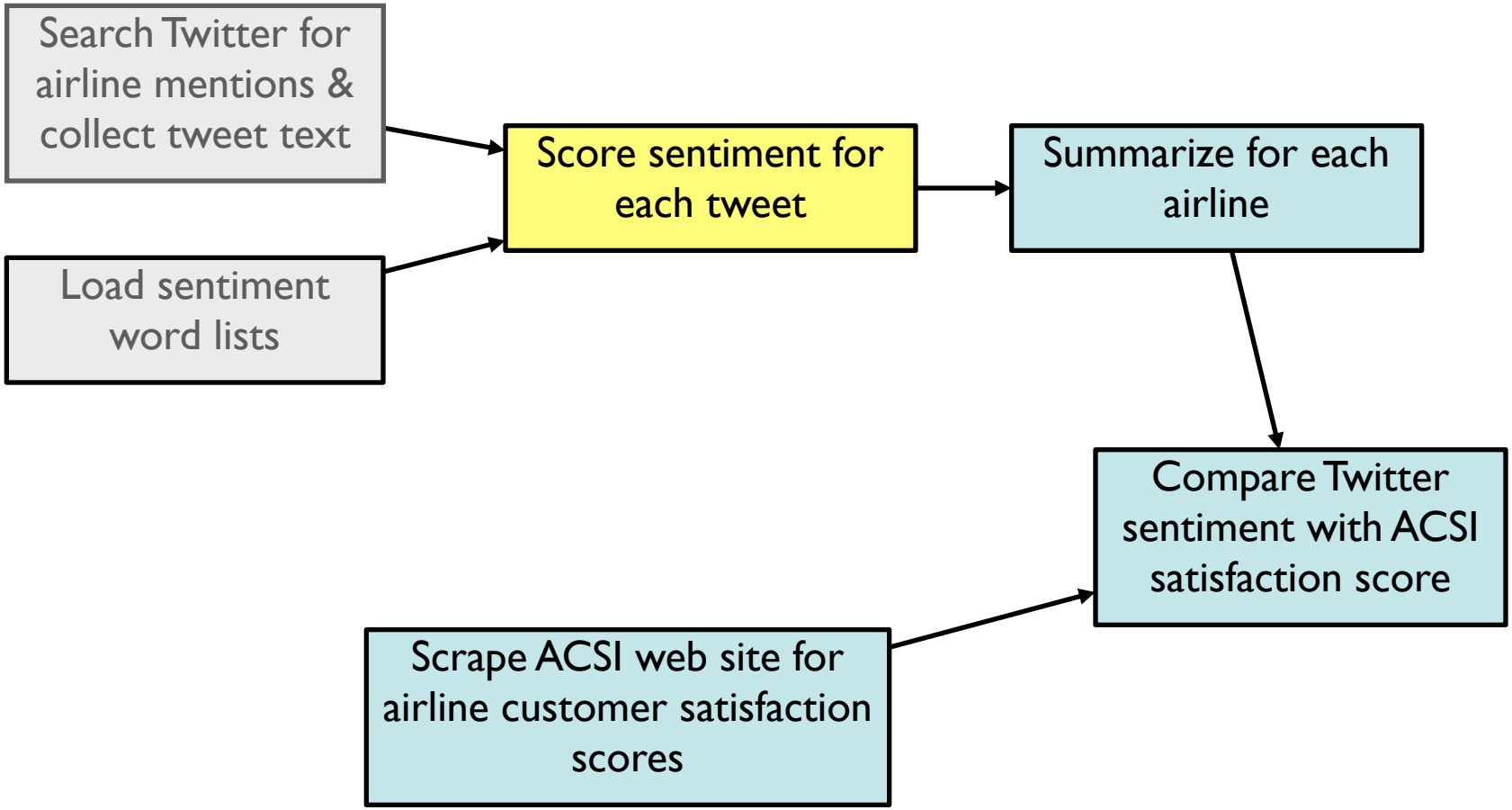
```
> hu.liu.pos = scan('../data/opinion-lexicon-English/positive-words.txt', what='character', comment.char=';')

> hu.liu.neg = scan('../data/opinion-lexicon-English/negative-words.txt', what='character', comment.char=';')
```

3. Add a few industry-specific and/or especially emphatic terms:

```
> pos.words = c(hu.liu.pos, 'upgrade')
> neg.words = c(hu.liu.neg, 'wtf', 'wait',
  'waiting', 'epicfail', 'mechanical')
```

The c() function combines objects into vectors or lists

Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Algorithm sanity check

```
> sample = c("You're awesome and I love you",
    "I hate and hate and hate. So angry. Die!",
    "Impressed and amazed: you are peerless in your achievement of
    unparalleled mediocrity.")
> result = score.sentiment(sample, pos.words, neg.words)
> class(result)
[1] "data.frame"
> result$score
[1]  2 -5  4
```

data.frames hold tabular data so they consist of columns & rows which can be accessed by name or number.

Here, “score” is the name of a column.

So, not so good with sarcasm. Here are a couple of real tweets:

```
> score.sentiment(c("@Delta I'm going to need you to get it together.
Delay on tarmac, delayed connection, crazy gate changes... #annoyed",
"Surprised and happy that @Delta helped me avoid the 3.5 hr layover I
was scheduled for.  Patient and helpful agents. #remarkable"),
pos.words, neg.words)$score
[1] -4  5
```

Accessing data.frames

Here's the data.frame just returned from `score.sentiment()`:

```
> result
 score                                                                  text
1     2                                                You're awesome and I love you
2    -5                                          I hate and hate and hate. So angry. Die!
3     4 Impressed and amazed: you are peerless in your achievement of unparalleled mediocrity.
```

Elements can be accessed by name or position, and positions can be ranges:

```
> result[1,1]
[1] 2
> result[1,'score']
[1] 2
> result[1:2, 'score']
[1]  2 -5
> result[c(1,3), 'score']
[1] 2 4
> result[,'score']
[1]  2 -5  4
```

Score the tweets

To score all of the Delta tweets, just feed their text into `score.sentiment()`:

```
> delta.scores = score.sentiment(delta.text, pos.words,
neg.words, .progress='text')
|==================================================| 100%
```

Progress bar provided by plyr

Let's add two new columns to identify the airline for when we combine all the scores later:

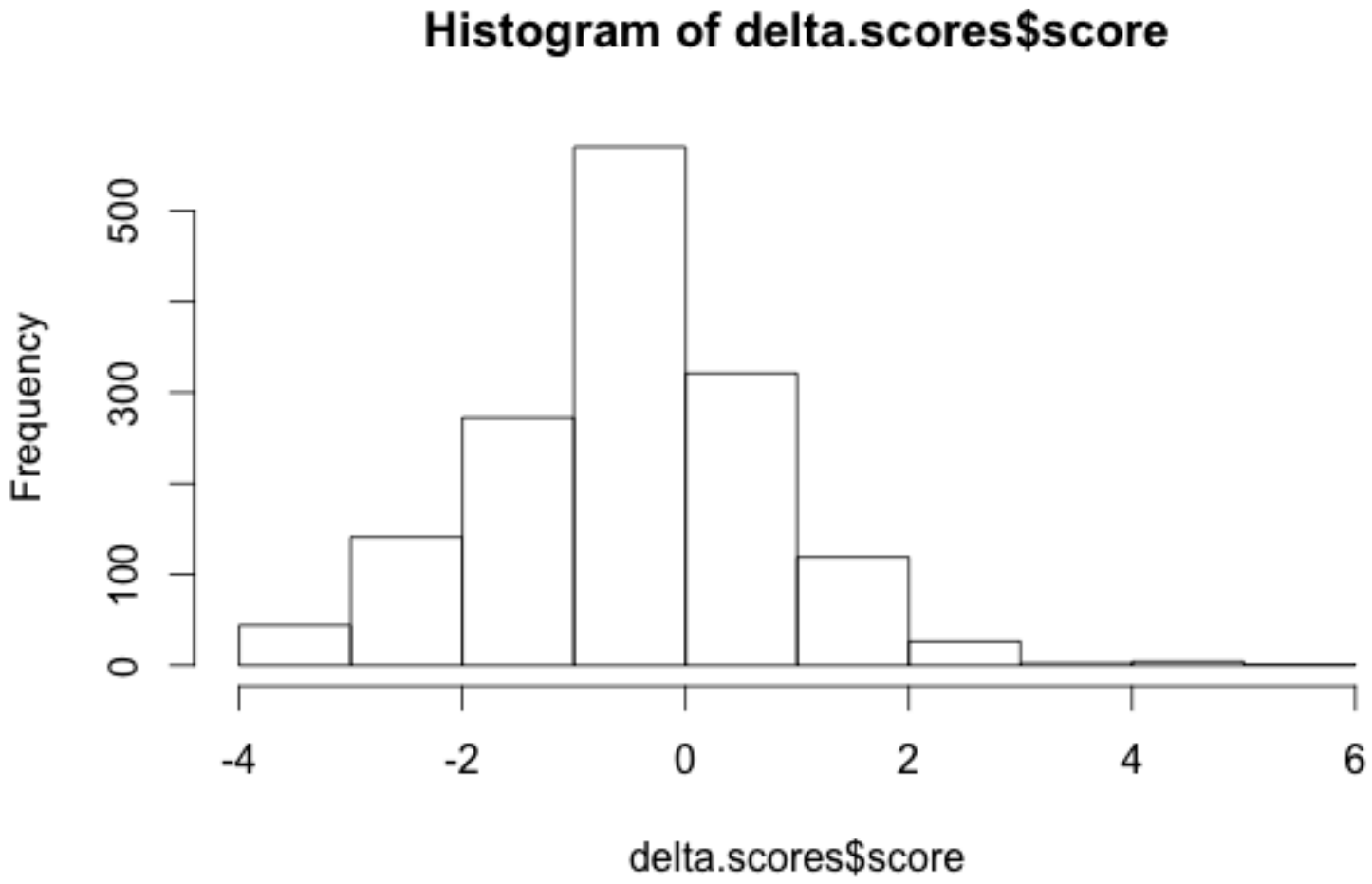
```
> delta.scores$airline = 'Delta'
> delta.scores$code = 'DL'
```

	score	text	airline	code
1	1	I am ready to head home. Inshallah will try to get on the earlier flight to Fresno. @Delta @DeltaAssist	Delta	DL
2	0	@Delta Releases 2010 Corporate Responsibility Report - @PRNewswire (press release) : http://tinyurl.com/64mz3oh	Delta	DL
3	1	Another week, another upgrade! Thanks @Delta!	Delta	DL
4	0	I'm not able to check in or select a seat for flight DL223/KL6023 to Seattle tomorrow. Help? @KLM @delta	Delta	DL
5	-3	In my boredom of waiting realized @deltaairlines is now @delta seriously..... Stil waiting and your not even unloading status yet	Delta	DL
6	1	Hmmm... I just got 'upgraded' from my reserved exit row seat to a knee banger. ATL-PVD just got longer. What gives @Delta?	Delta	DL
7	0	@Delta 7 days I'm trying to book a flight with you. Still w/o success. Starting to get really upset. What can I do?	Delta	DL
8	-2	its amazing how horrible @Delta there service is horrendous and their staff unprofessional! #angry	Delta	DL
9	0	With @DeltaAssist, do you believe @Delta has seen greater returns through #SM efforts? http://bit.ly/kXn9qZ	Delta	DL
10	-3	@AmericanAir app froze at TSA Checkpoint. Had to leave line to get paper boarding pass; never have this problem with @Delta app #fail	Delta	DL

Plot Delta's score distribution

R's built-in `hist()` function will create and plot histograms of your data:

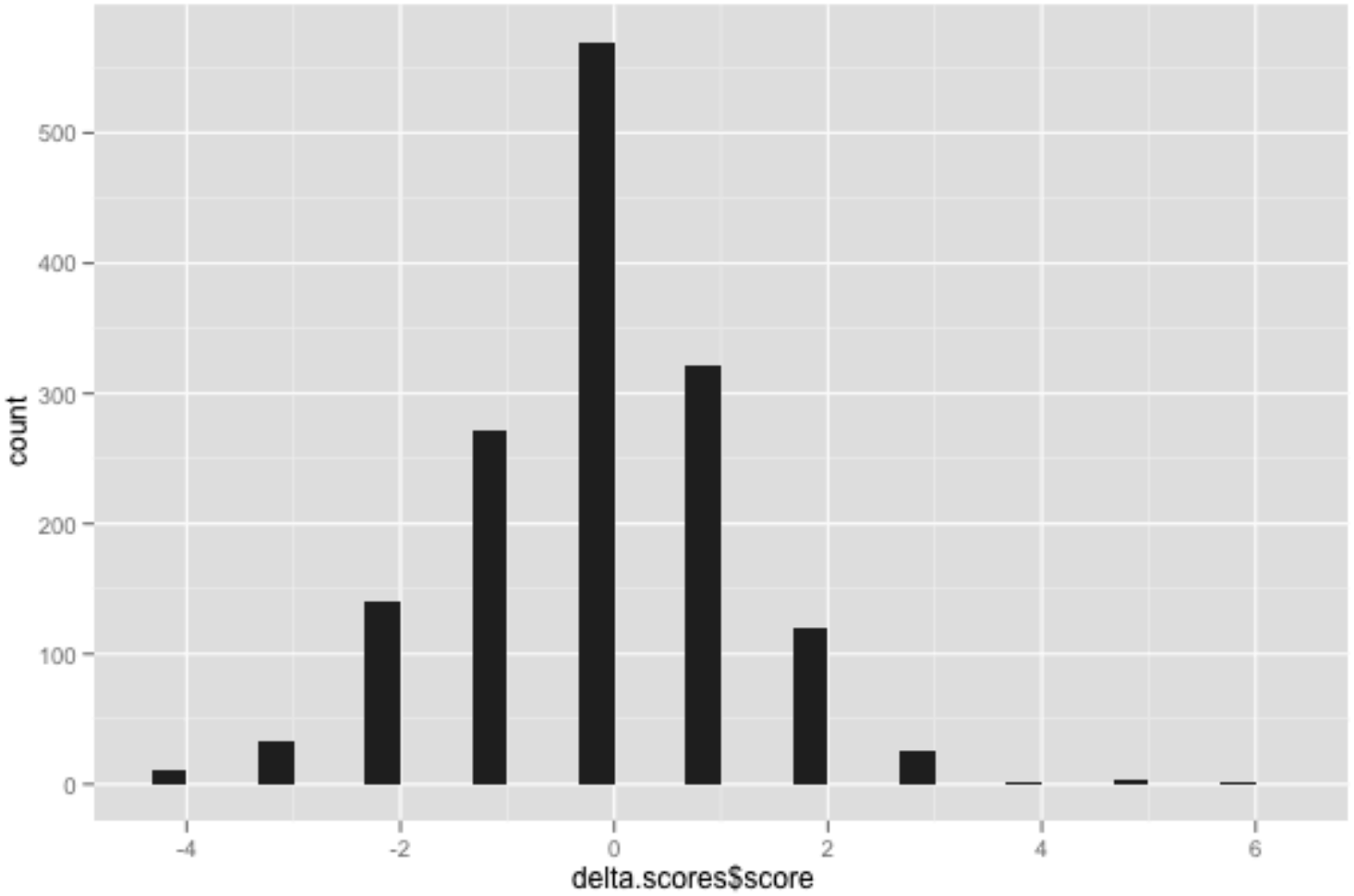
```
> hist(delta.scores$score)
```



The ggplot2 alternative

ggplot2 is an alternative graphics package which generates more refined graphics:

```
> qplot(delta.scores$score)
```



Lather. Rinse. Repeat

To see how the other airlines fare, collect & score tweets for other airlines.

Then combine all the results into a single “all.scores” data.frame:

```
> all.scores = rbind( american.scores, continental.scores, delta.scores,
jetblue.scores, southwest.scores, united.scores, us.scores )
```

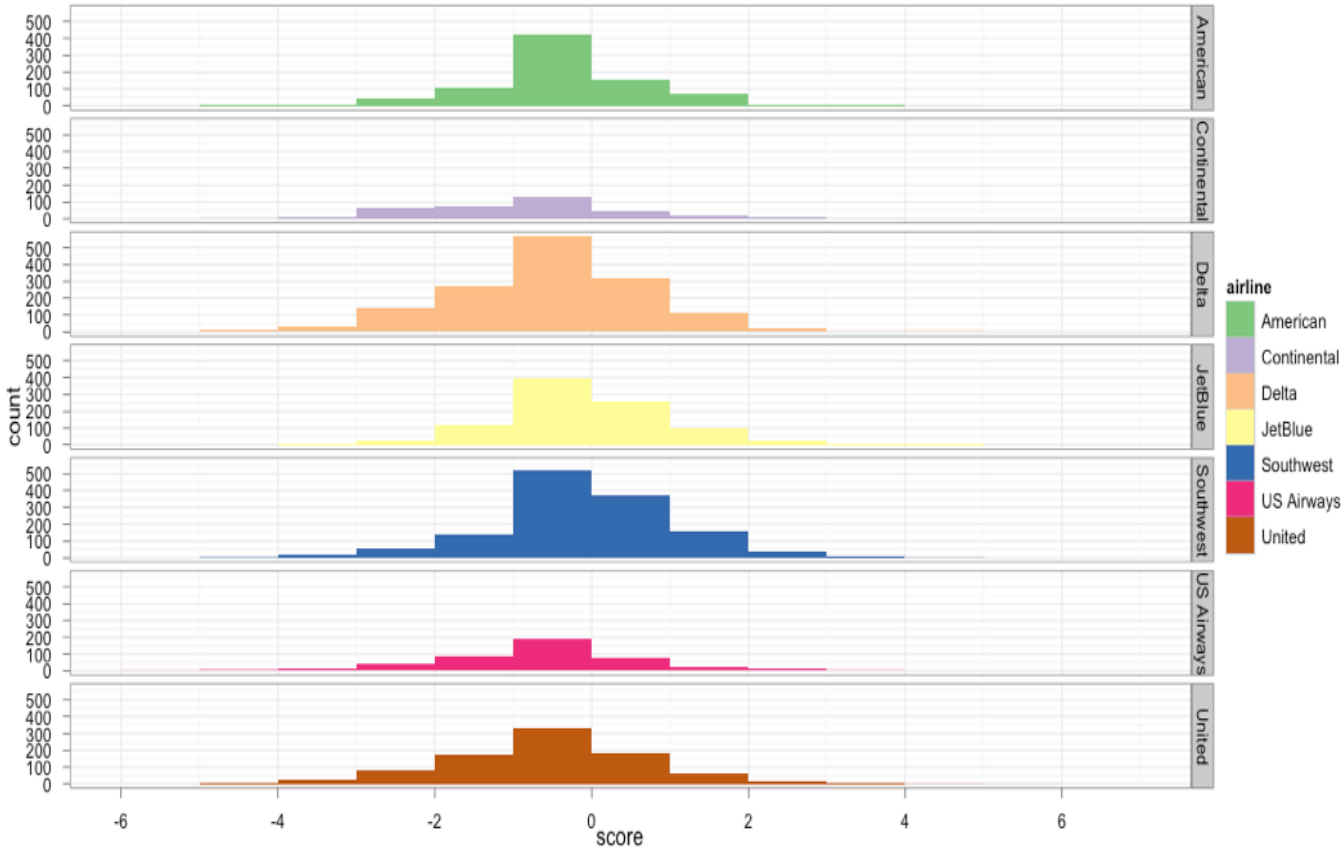
`rbind()` combines rows from data.frames, arrays, and matrices

Compare score distributions

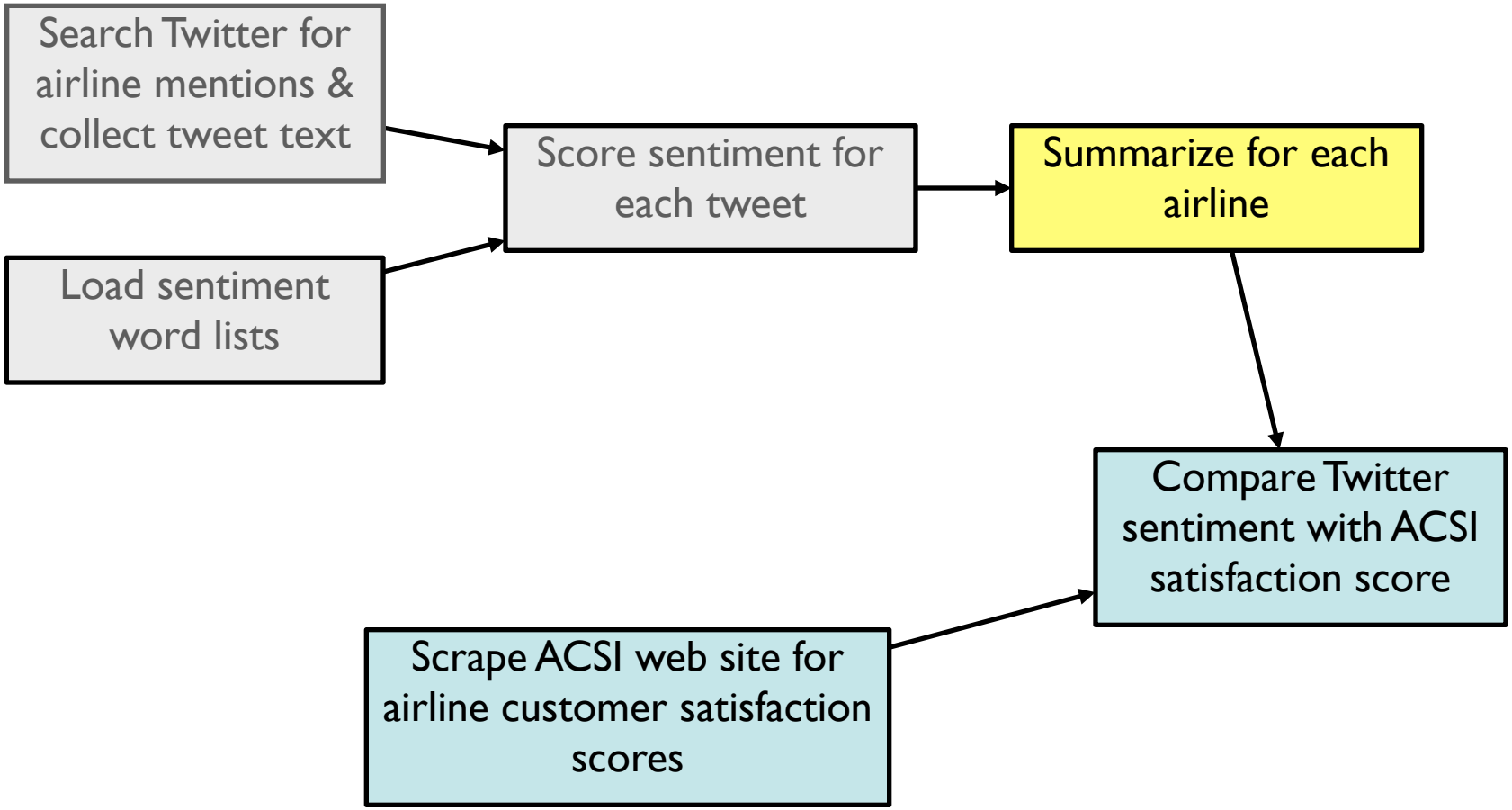
ggplot2 implements “grammar of graphics”, building plots in layers:

```
> ggplot(data=all.scores) + # ggplot works on data.frames, always
    geom_bar(mapping=aes(x=score, fill=airline), binwidth=1) +
    facet_grid(airline~.) + # make a separate plot for each airline
    theme_bw() + scale_fill_brewer() # plain display, nicer colors
```

ggplot2’s faceting capability makes it easy to generate the same graph for different values of a variable, in this case “airline”.



Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Ignore the middle

Let's focus on very negative (<-2) and positive (>2) tweets:

```
> all.scores$very.pos = as.numeric( all.scores$score >= 2 )
> all.scores$very.neg = as.numeric( all.scores$score <= -2 )
```

For each airline (airline + code), let's use the ratio of very positive to very negative tweets as the overall sentiment score for each airline:

```
> twitter.df = ddply(all.scores, c('airline', 'code'), summarise,
pos.count = sum( very.pos ), neg.count = sum( very.neg ) )
> twitter.df$all.count = twitter.df$pos.count + twitter.df$neg.count
> twitter.df$score = round( 100 * twitter.df$pos.count /
                  twitter.df$all.count )
```

Sort with `orderBy()` from the doBy package:

```
> orderBy(~-score, twitter.df)
```

	airline	code	pos.count	neg.count	all.count	score
1	JetBlue	B6	146	28	174	84
2	Southwest	WN	207	72	279	74
3	American	AA	80	57	137	58
4	Delta	DL	152	185	337	45
5	United	UA	82	102	184	45
6	US Airways	US	38	62	100	38
7	Continental	CO	22	68	90	24

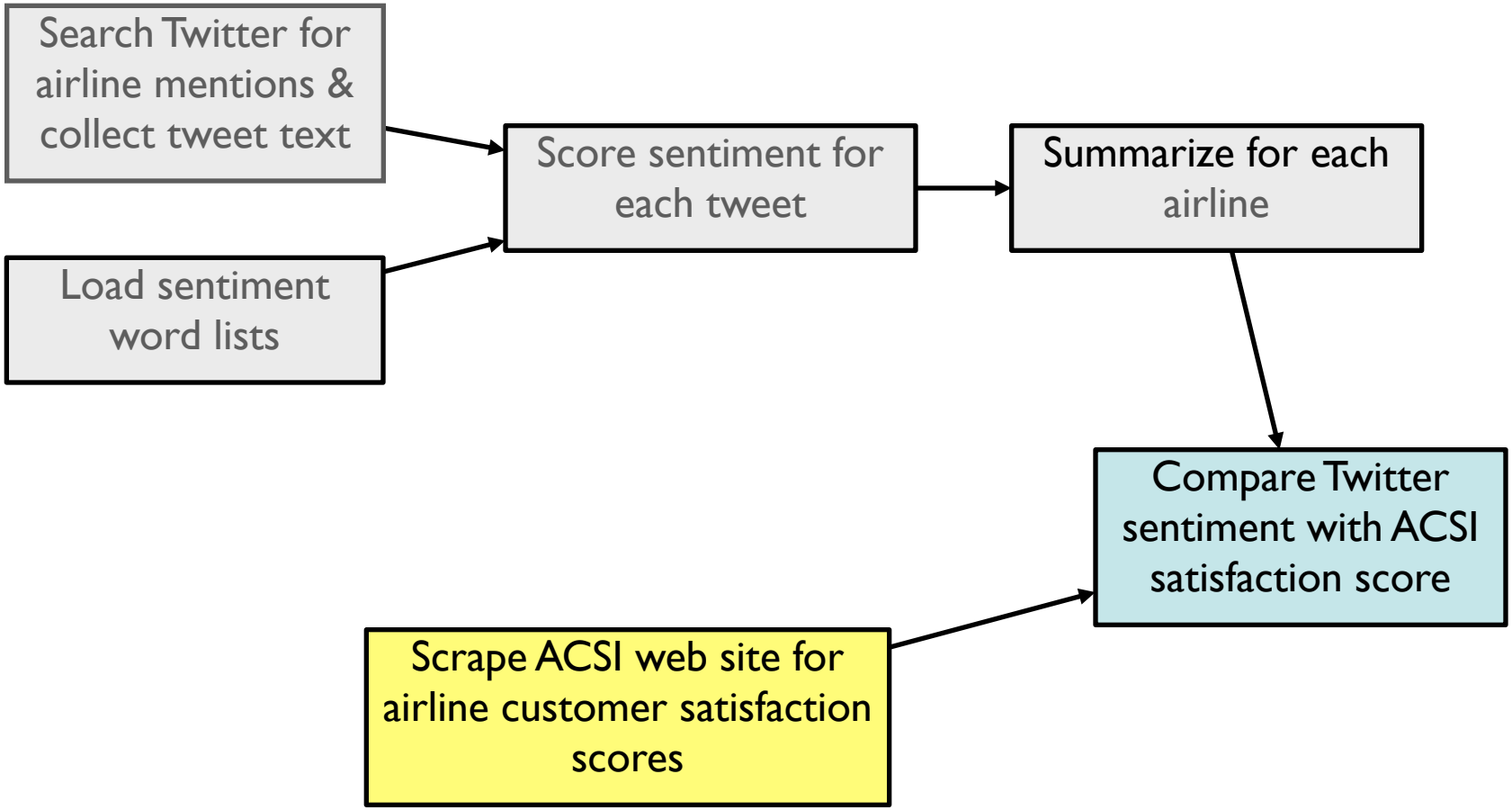
Any relation to ACSI's airline scores?

Airlines

	Base-line	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	Previous Year % Change	First Year % Change
Southwest Airlines	78	76	76	76	74	72	70	70	74	75	73	74	74	76	79	81	79		-2.5	1.3
All Others	NM	70	74	70	62	67	63	64	72	74	73	74	74	75	75	77	75		-2.6	7.1
Continental Airlines	67	64	66	64	66	64	62	67	68	68	67	70	67	69	62	68	71		4.4	6.0
Airlines	**72**	**69**	**69**	**67**	**65**	**63**	**63**	**61**	**66**	**67**	**66**	**66**	**65**	**63**	**62**	**64**	**66**		**3.1**	**-8.3**
American Airlines	70	71	71	62	67	64	63	62	63	67	66	64	62	60	62	60	63		5.0	-10.0
Delta Air Lines (Delta)	77	72	67	69	65	68	66	61	66	67	67	65	64	59	60	64	62		-3.1	-19.5
US Airways	72	67	66	68	65	61	62	60	63	64	62	57	62	61	54	59	62		5.1	-13.9
Northwest Airlines (Delta)	69	71	67	64	63	53	62	56	65	64	64	64	61	61	57	57	61		7.0	-11.6
United Airlines	71	67	70	68	65	62	62	59	64	63	64	61	63	56	56	56	60		7.1	-15.5

http://www.theacsi.org/index.php?option=com_content&view=article&id=147&catid=&Itemid=212&i=Airlines

Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Scrape, don't type

XML package provides amazing `readHTMLtable()` function:

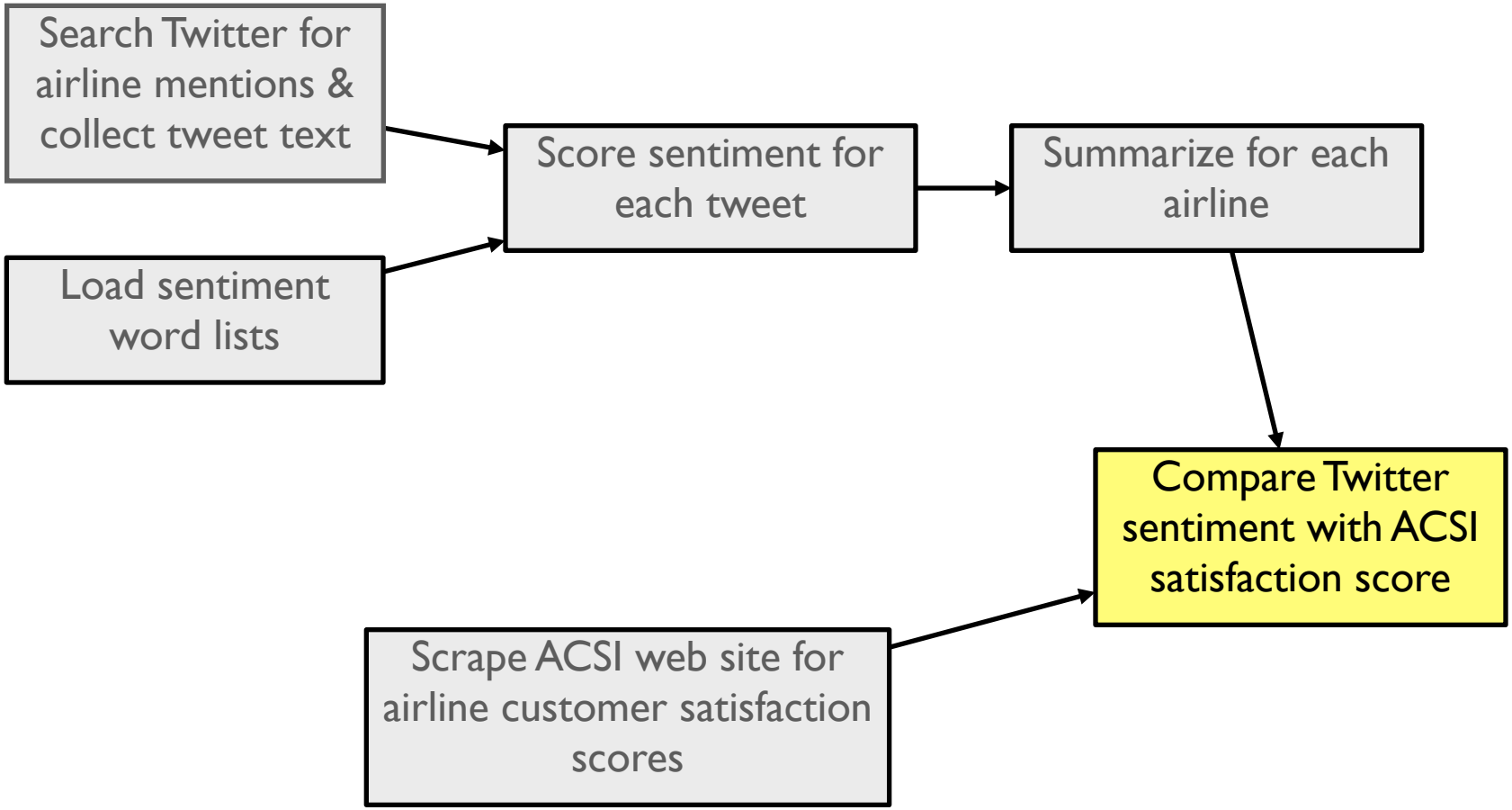
```
> library(XML)
> acsi.url = 'http://www.theacsi.org/index.php?option=com_content&view=article&id=147&catid=&Itemid=212&i=Airlines'
> acsi.df = readHTMLTable(acsi.url, header=T, which=1, stringsAsFactors=F)
> # only keep column #1 (name) and #18 (2010 score)
> acsi.df = acsi.df[,c(1,18)]
> head(acsi.df,1)
                    10
1 Southwest Airlines 79
```

Well, typing metadata is OK, I guess... clean up column names, etc:

```
> colnames(acsi.df) = c('airline', 'score')
> acsi.df$code = c('WN', NA, 'CO', NA, 'AA', 'DL',
                   'US', 'NW', 'UA')
> acsi.df$score = as.numeric(acsi.df$score)
```

NA (as in "n/a") is supported as a valid value everywhere in R.

Game Plan


Search Twitter for airline mentions & collect tweet text
Load sentiment word lists
Score sentiment for each tweet
Summarize for each airline
Compare Twitter sentiment with ACSI satisfaction score
Scrape ACSI web site for airline customer satisfaction scores

Join and compare

`merge()` joins two data.frames by the specified "by=" fields. You can specify 'suffixes' to rename conflicting column names:

```
> compare.df = merge(twitter.df, acsi.df, by='code',
    suffixes=c('.twitter', '.acsi'))
```

	code	airline.twitter	pos.count	neg.count	all.count	score.twitter	airline.acsi	score.acsi
1	AA	American	80	57	137	58	American Airlines	63
2	CO	Continental	22	68	90	24	Continental Airlines	71
3	DL	Delta	152	185	337	45	Delta Air Lines (Delta)	62
4	UA	United	82	102	184	45	United Airlines	60
5	US	US Airways	38	62	100	38	US Airways	62
6	WN	Southwest	207	72	279	74	Southwest Airlines	79

Unless you specify "all=T", non-matching rows are dropped (like a SQL INNER JOIN), and that's what happened to top scoring JetBlue.

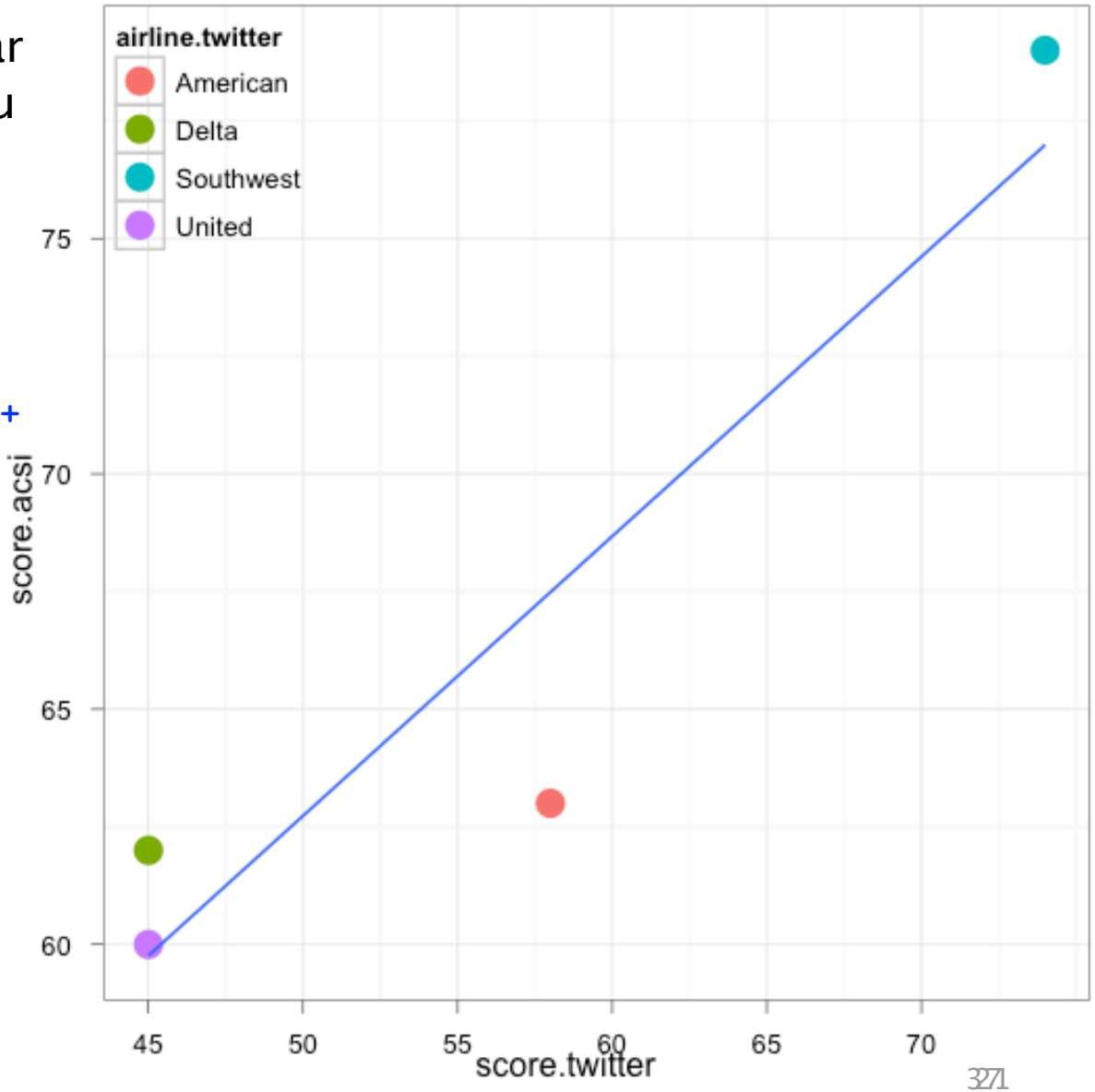
With a very low score, and low traffic to boot, soon-to-disappear Continental looks like an outlier. Let's exclude:

```
> compare.df = subset(compare.df, all.count > 100)
```

an actual result!

ggplot will even run `lm()` linear (and other) regressions for you with its geom_smooth() layer:

```
> ggplot( compare.df ) +
geom_point(aes(x=score.twitter,
y=score.acsi,
color=airline.twitter), size=5) +
geom_smooth(aes(x=score.twitter,
y=score.acsi, group=1), se=F,
method="lm") +
theme_bw() +
opts(legend.position=c(0.2,
0.85))
```



http://www.despair.com/cudi.html

R code for example scoring function

```
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
	require(plyr)
	require(stringr)

	# we got a vector of sentences. plyr will handle a list or a vector as an "l" for us
	# we want a simple array of scores back, so we use "l" + "a" + "ply" = laply:
	scores = laply(sentences, function(sentence, pos.words, neg.words) {

		# clean up sentences with R's regex-driven global substitute, gsub():
		sentence = gsub('[[:punct:]]', '', sentence)
		sentence = gsub('[[:cntrl:]]', '', sentence)
		sentence = gsub('\\d+', '', sentence)
		# and convert to lower case:
		sentence = tolower(sentence)

		# split into words. str_split is in the stringr package
		word.list = str_split(sentence, '\\s+')
		# sometimes a list() is one level of hierarchy too much
		words = unlist(word.list)

		# compare our words to the dictionaries of positive & negative terms
		pos.matches = match(words, pos.words)
		neg.matches = match(words, neg.words)

		# match() returns the position of the matched term or NA
		# we just want a TRUE/FALSE:
		pos.matches = !is.na(pos.matches)
		neg.matches = !is.na(neg.matches)

		# and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
		score = sum(pos.matches) - sum(neg.matches)

		return(score)
	}, pos.words, neg.words, .progress=.progress )

	scores.df = data.frame(score=scores, text=sentences)
	return(scores.df)
}
```