Designing Exploratory Visualization Tools

Basic charts for EDA

Table

Bar graph, dot plot

Histogram, frequency polygon, strip plot

Line graph

Scatter plot, bubble (size for third variable)

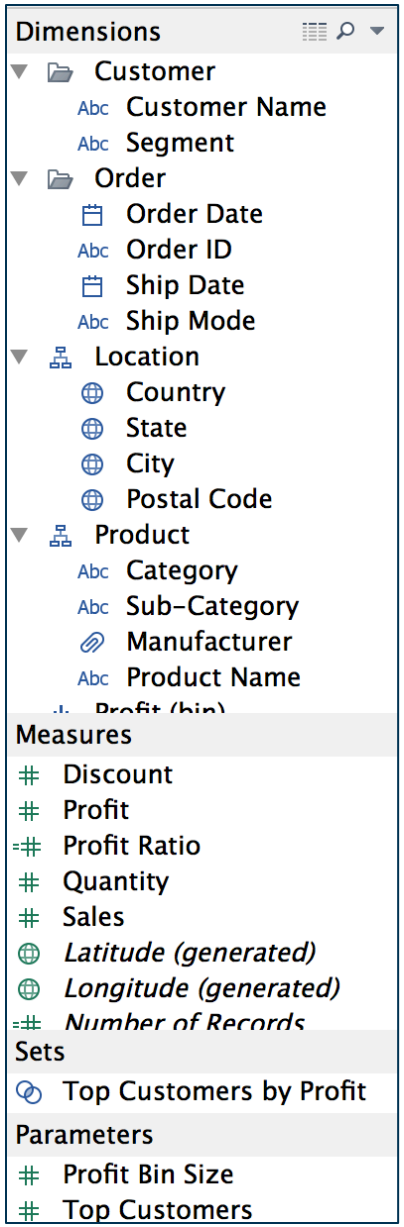
Box plot

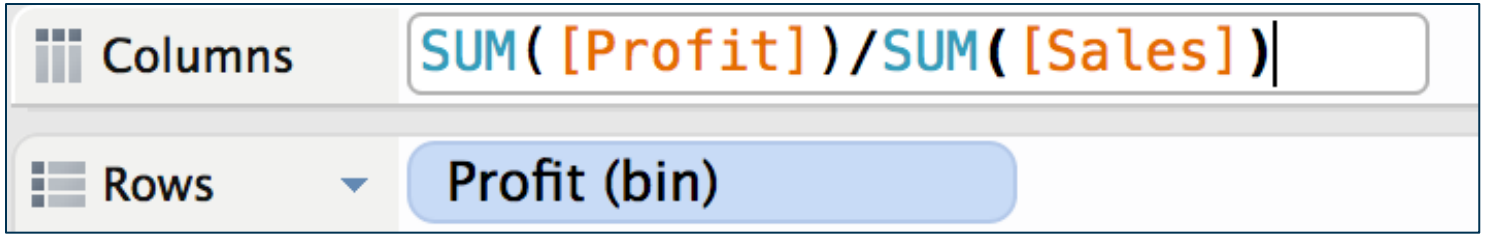
Heatmap matrix

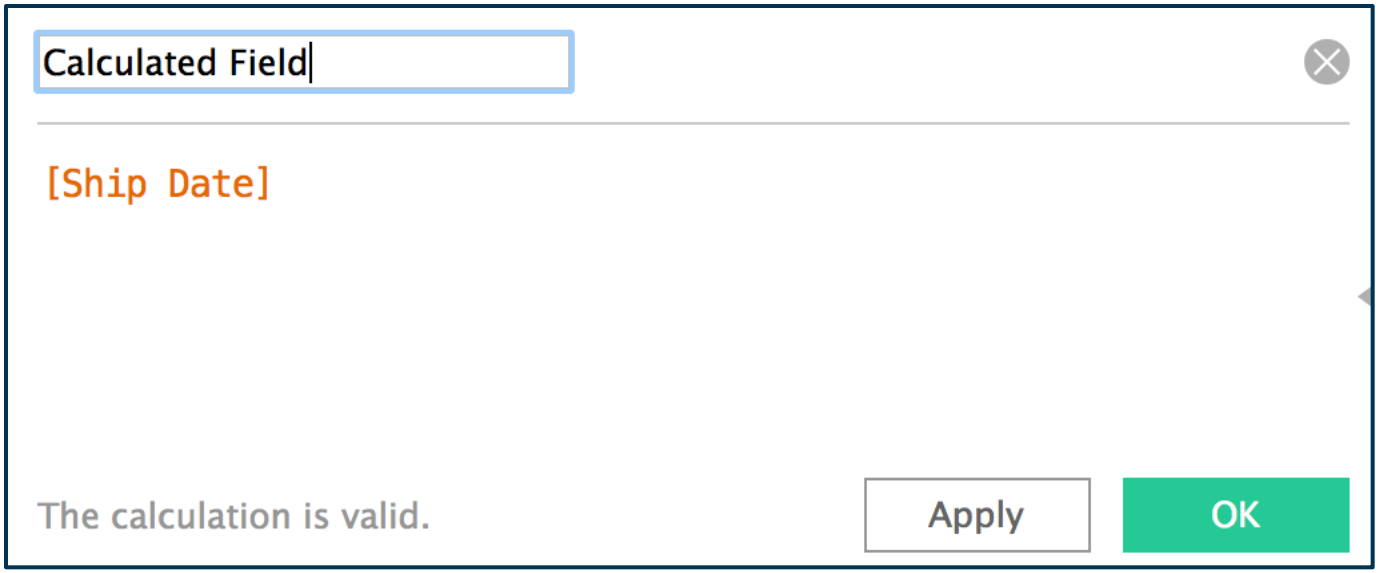
Bar/line or bar/dot combination

What makes for a good visual EDA tool?

Seamless data interaction


Dimensions
Customer
Customer Name
Segment
Order
Order Date
Order ID
Ship Date
Ship Mode
Location
Country
State
City
Postal Code
Product
Category
Sub-Category
Manufacturer
Product Name
Measures
Discount
Profit
Profit Ratio
Quantity
Sales
Latitude (generated)
Longitude (generated)
Sets
Top Customers by Profit
Parameters
Profit Bin Size
Top Customers


Columns
SUM([Profit])/SUM([Sales])
Rows
Profit (bin)

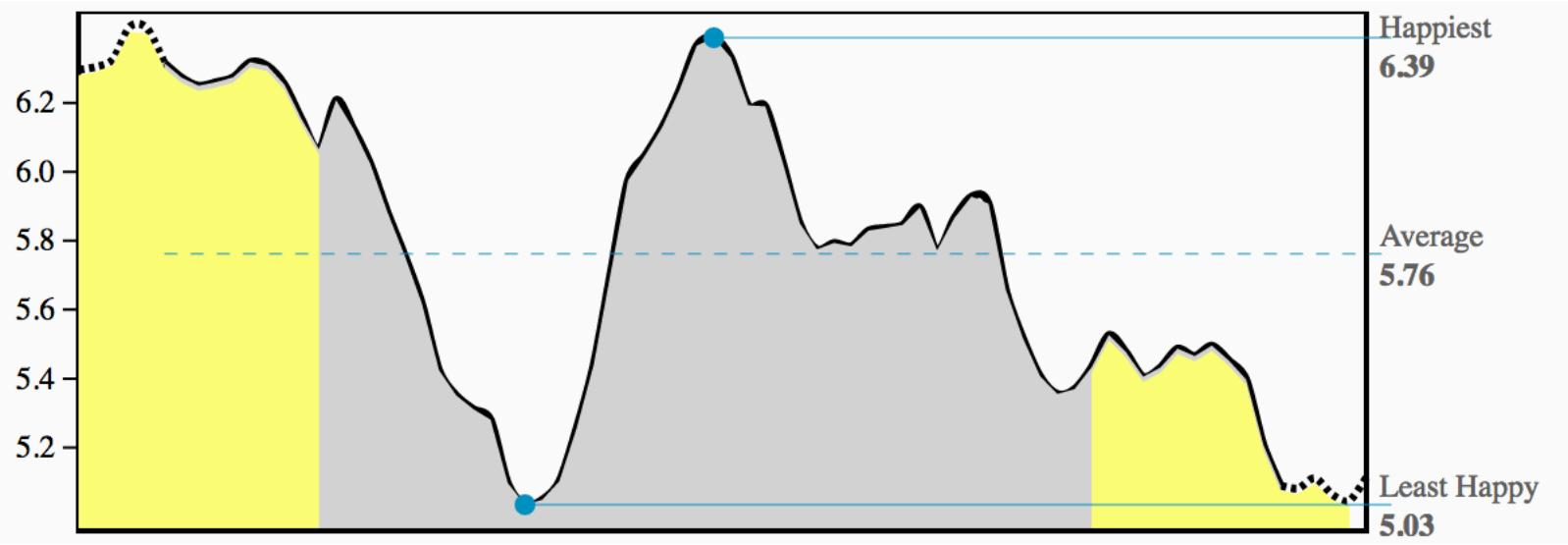

Calculated Field
[Ship Date]
The calculation is valid.
Apply
OK

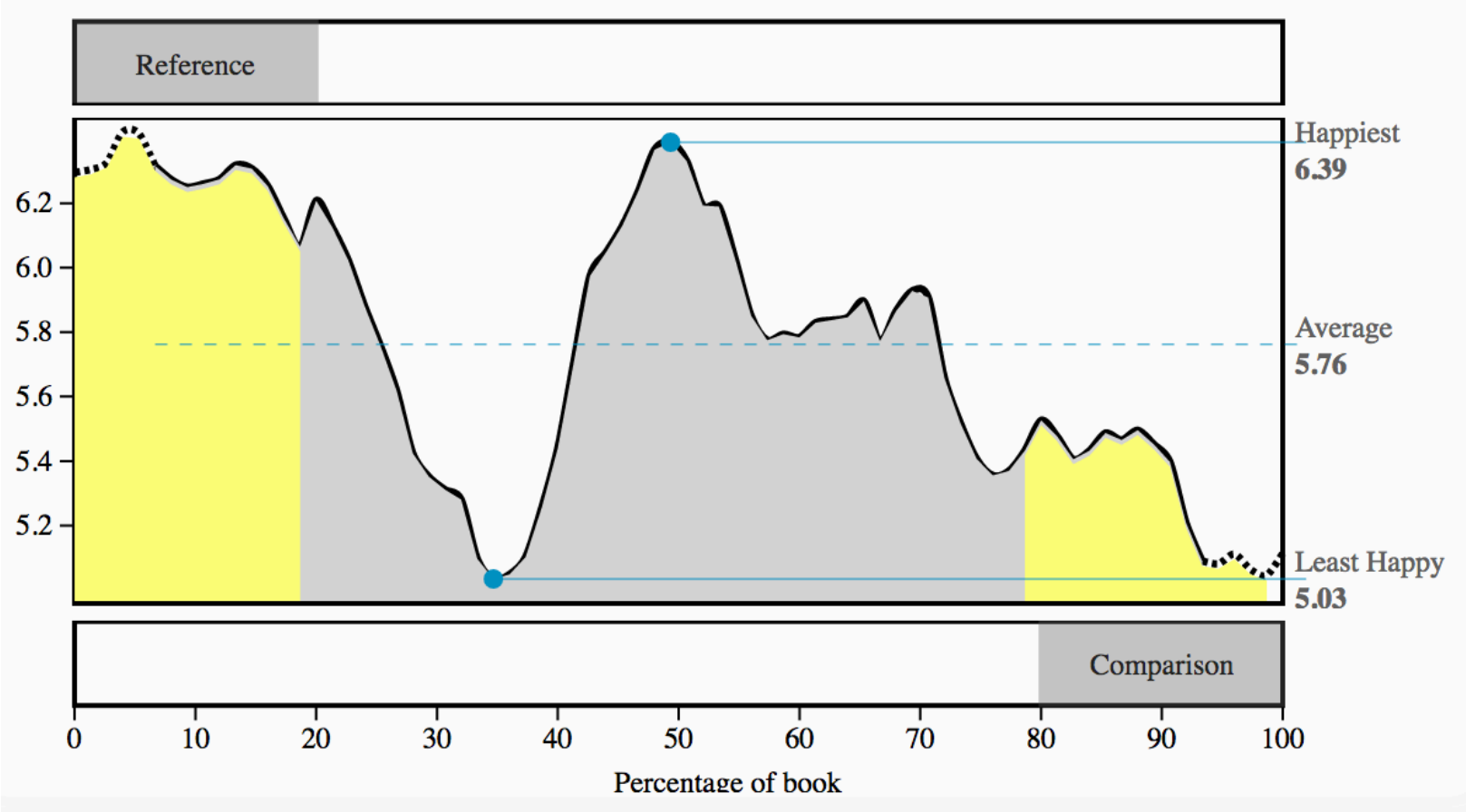
	user_id	text	diag_date	from_diag	created_date	target
158437	2906418756	RT @karimaro11: 先月から家に居るコマちゃん。30日間を30秒間にまとめてみま...	2014-06-11	582.0	2016-01-14	1
24971	86740068	Home made fried oreos. Yum! http://yfrog.com/k...	None	NaN	2011-08-07	0
122398	1389531	Marti Belle and Tina San Antonio back when the...	2015-11-10	60.0	2016-01-09	1
84403	248388481	HOLY FUCKING SHIT!!!!!!!!!!! #HannibalFinale @...	None	NaN	2014-05-24	0
47950	45013513	MobaXTerm is pretty much the best thing ever.	None	NaN	2015-04-26	0

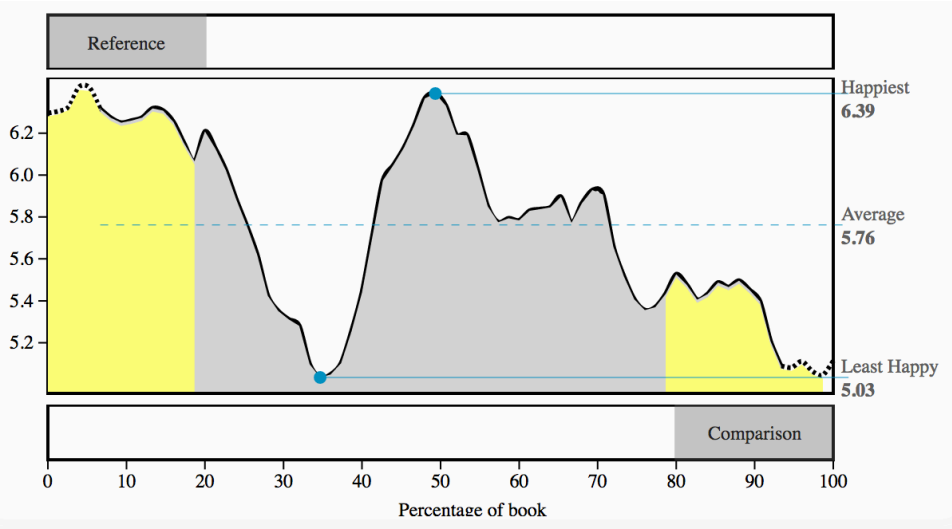
Data transformation operations (e.g., `df.groupByKey()`).

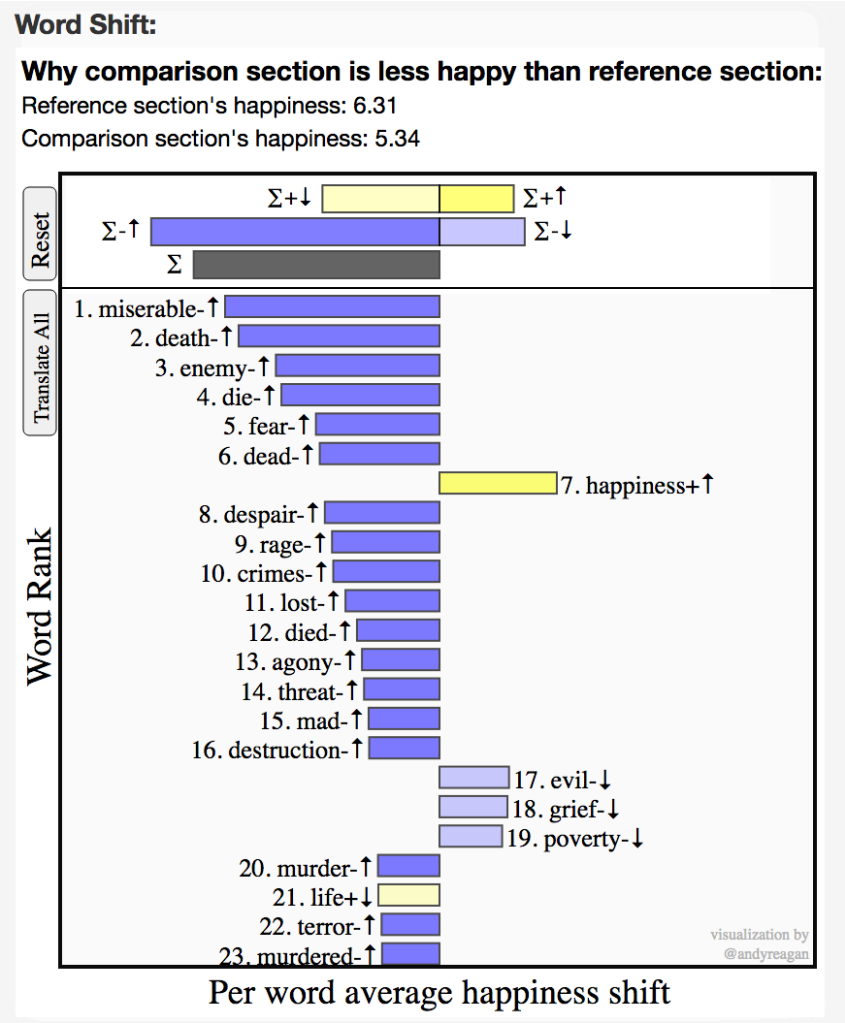
```
collapse_count = 0
# first thing, need to collapse everything onto user-days
# i'm going to hack the dataframe object to do this
target_users = dict()
for df_id,r in tqdm(q.iterrows()):
    if r.user_id in target_users:
        if r.from_diag in target_users[r.user_id]:
            target_users[r.user_id][r.from_diag] += " "+r.text
            collapse_count += 1
        else:
            target_users[r.user_id][r.from_diag] = r.text
    else:
        target_users[r.user_id] = dict()
        target_users[r.user_id][r.from_diag] = r.text
empty_count = 0
# turn each observation into a sparse, normalized word vector
for user in tqdm(target_users):
    for date in target_users[user]:
        target_users[user][date] = my_LabMT.wordVecify(dictify(listify(target_users[user][date])))
        # print(target_users[user][date].sum())
        # print(target_users[user][date])
        # print(target_users[user][date]/target_users[user][date].sum())
        if target_users[user][date].sum() != 0:
            target_users[user][date] = target_users[user][date]*10/(target_users[user][date].sum())
        else:
            empty_count+=1
        target_users[user][date] = csr_matrix(target_users[user][date])
control_users = dict()
collapse_count_control = 0
for df_id,r in tqdm(p.iterrows()):
    if r.user_id in control_users:
        if r.created_date in control_users[r.user_id]:
            control_users[r.user_id][r.created_date] += " "+r.text
            collapse_count_control += 1
        else:
            control_users[r.user_id][r.created_date] = r.text
    else:
        control_users[r.user_id] = dict()
        control_users[r.user_id][r.created_date] = r.text
for user in tqdm(control_users):
    for date in control_users[user]:
        control_users[user][date] = my_LabMT.wordVecify(dictify(listify(control_users[user][date])))
        if control_users[user][date].sum() != 0:
            control_users[user][date] = control_users[user][date]*10/control_users[user][date].sum()
        control_users[user][date] = csr_matrix(control_users[user][date])
```

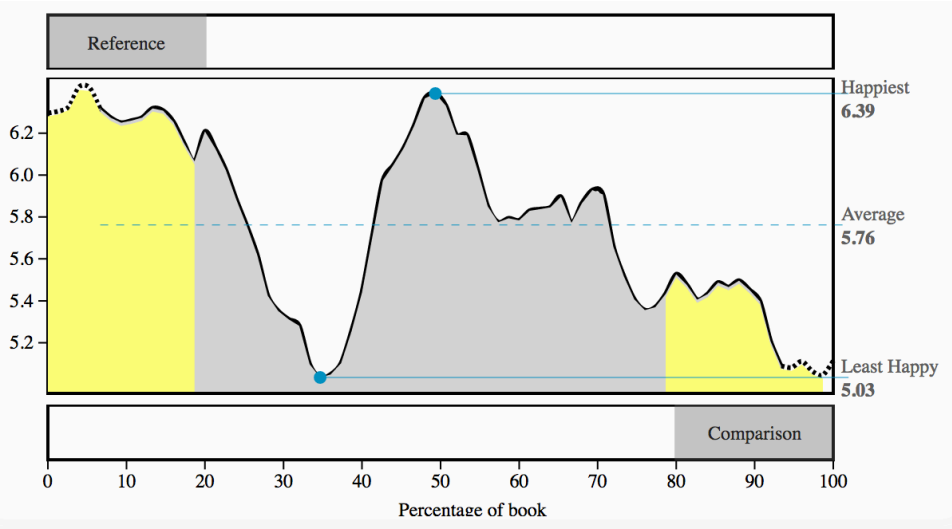
Rich data comparison

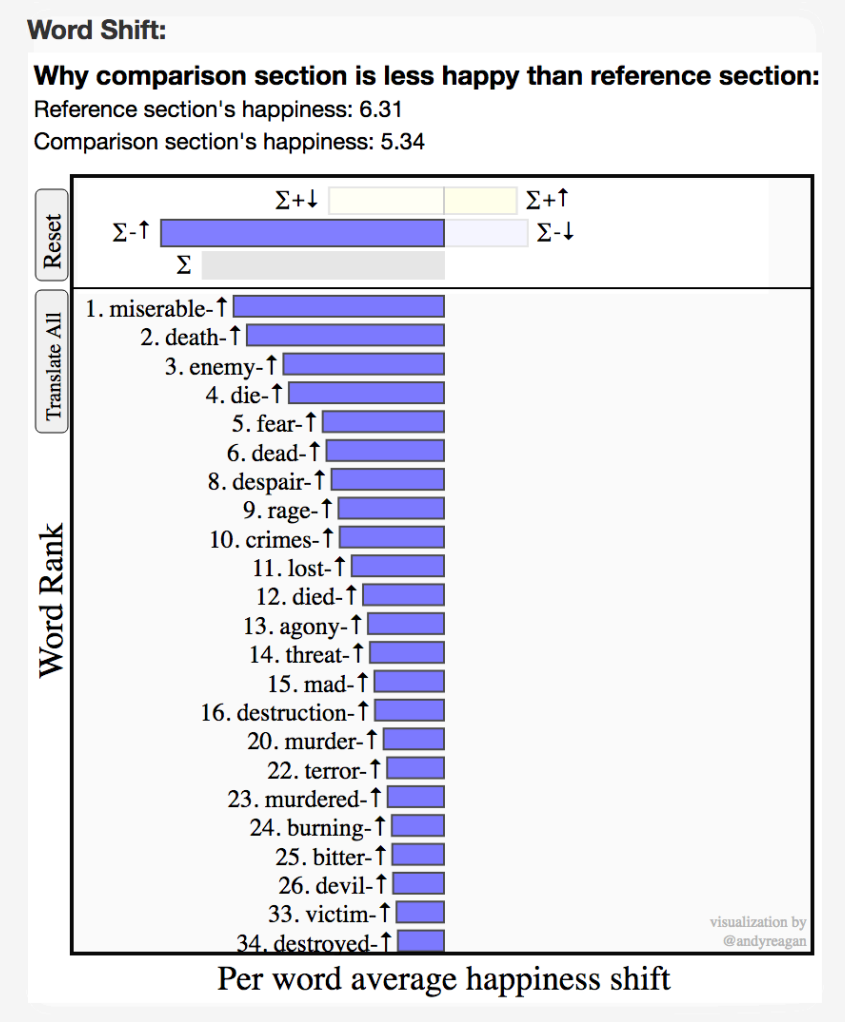

6.2
6.0
5.8
5.6
5.4
5.2
Happiest
6.39
Average
5.76
Least Happy
5.03


Reference
Happiest
6.39
Average
5.76
Least Happy
5.03
Comparison
6.2
6.0
5.8
5.6
5.4
5.2
0
10
20
30
40
50
60
70
80
90
100
Percentage of book

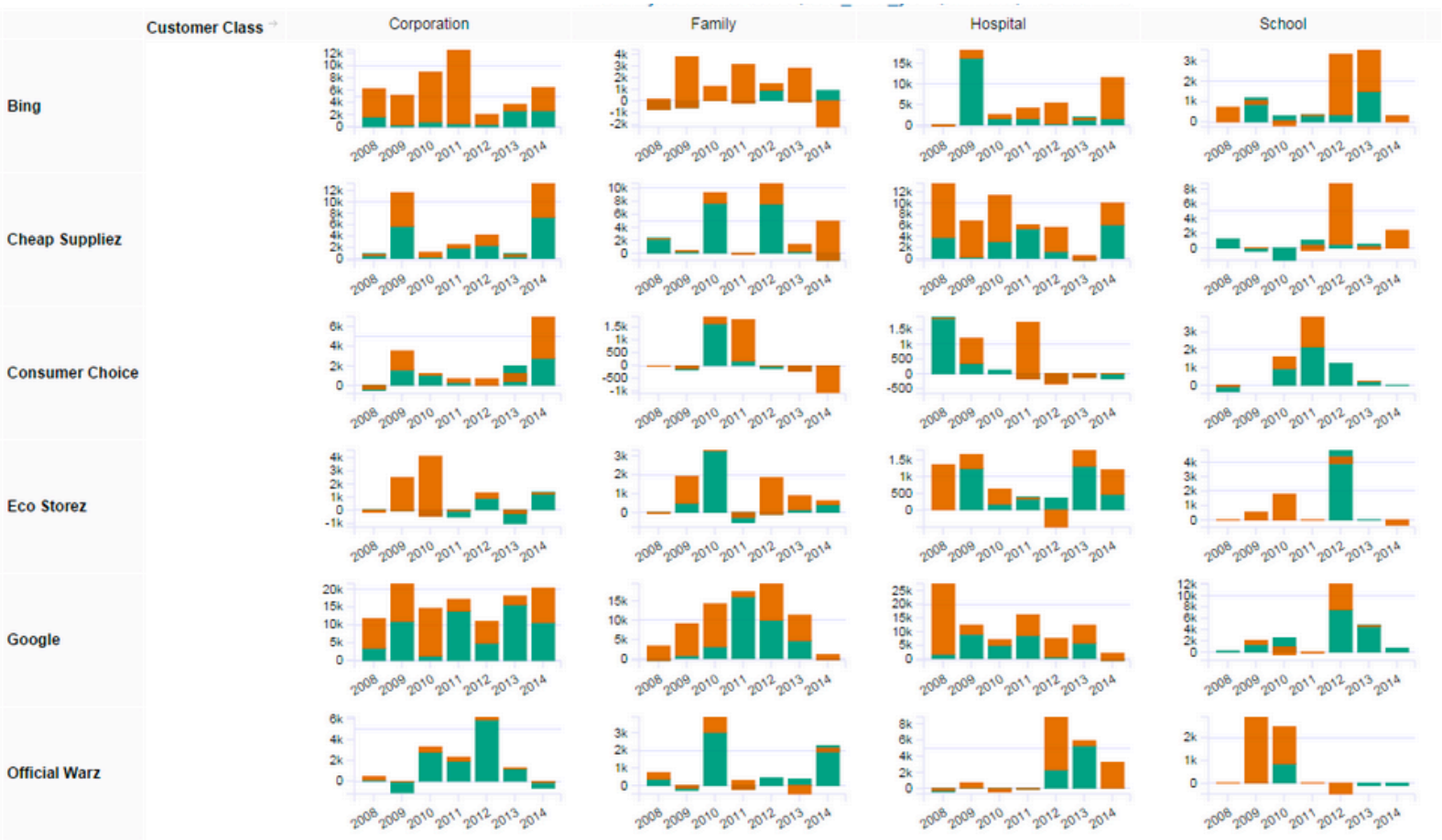

Reference
Happiest
6.39
Average
5.76
Least Happy
5.03
Comparison
6.2
6.0
5.8
5.6
5.4
5.2
0
10
20
30
40
50
60
70
80
90
100
Percentage of book


Word Shift:
Why comparison section is less happy than reference section:
Reference section's happiness: 6.31
Comparison section's happiness: 5.34
Reset
Translate All
Σ+↓
Σ+↑
Σ-↑
Σ-↓
Σ
1. miserable-↑
2. death-↑
3. enemy-↑
4. die-↑
5. fear-↑
6. dead-↑
7. happiness+↑
8. despair-↑
9. rage-↑
10. crimes-↑
11. lost-↑
12. died-↑
13. agony-↑
14. threat-↑
15. mad-↑
16. destruction-↑
17. evil-↓
18. grief-↓
19. poverty-↓
20. murder-↑
21. life+↓
22. terror-↑
23. murdered-↑
Word Rank
visualization by
@andyreagan
Per word average happiness shift


Reference
Comparison
Happiest
6.39
Average
5.76
Least Happy
5.03
6.2
6.0
5.8
5.6
5.4
5.2
0
10
20
30
40
50
60
70
80
90
100
Percentage of book


Word Shift:
Why comparison section is less happy than reference section:
Reference section's happiness: 6.31
Comparison section's happiness: 5.34
Reset
Translate All
Σ+↓
Σ+↑
Σ-↑
Σ-↓
Σ
Word Rank
1. miserable-↑
2. death-↑
3. enemy-↑
4. die-↑
5. fear-↑
6. dead-↑
8. despair-↑
9. rage-↑
10. crimes-↑
11. lost-↑
12. died-↑
13. agony-↑
14. threat-↑
15. mad-↑
16. destruction-↑
20. murder-↑
22. terror-↑
23. murdered-↑
24. burning-↑
25. bitter-↑
26. devil-↑
33. victim-↑
34. destroyed-↑
visualization by
@andyreagan
Per word average happiness shift

Multifaceted views


Customer Class
Corporation
Family
Hospital
School
Bing
Cheap Suppliez
Consumer Choice
Eco Storez
Google
Official Warz
2008
2009
2010
2011
2012
2013
2014

Integrated statistical calculations

Data access and integration

Berkeley SCHOOL OF INFORMATION