FOR ADDITIONAL THINGS SEE

HTTP://MCNUTT.IN/

GRAPHICS FROM NOUN PROJECT

HOW TO READ AN ACADEMIC PAPER

A NON-DEFINITIVE GUIDE

BY ANDREW MCNUTT

OR, WHAT I WISH I LEARNED IN COLLEGE

(INSTEAD OF JUST TAKING MATH AND PHYSICS CLASSES)

I read a lot of papers, such is the loathsome life of the PhD student. Unfortunately I foolishly squandered my first opportunity to learn how to academically read good and have had to develop these skills during grad school

I spent most of my college years insisting I didn't need to know how to read papers (or really to think critically at all) because scientists just did math. How wrong I was!

This zine will take you through my approach to reading papers. It mostly applies to reading Visualization and HCI papers, so your mileage may vary.

There are better guides on read gooding, but this documents what I wish I could tell myself 10 years ago.

ADDENDUM 1

Figure out when works for you to read

I find that I am most consistent with my reading when it's the first thing I do every day (some say ***"****if something matters to you, do it first thing in the morning****"***). I find if I read at night then it takes longer and I read in unnecessarily high depth.

ADDENDUM 2

Figure out what works for you to read

While some papers are really boring, not all of them are! If you are having a consistently hard time reading a particular kind of paper, that may be a sign that you find that type of paper boring. This is okay! You learned something about yourself. (You probably still gotta read them though.)

ADDENDUM 3

Don't read too deeply / don't read too shallowly

Some papers require great reflection and care, others require casual parses, others still require somewhere in between. I think one of the big skills in paper reading (as it is a skill, and not (as I have thought in the past) a talent) is to know when to read and how.

STEP 4. WRITE IT UP

You are probably going to forget the specifics of this paper in a few months. That's okay!

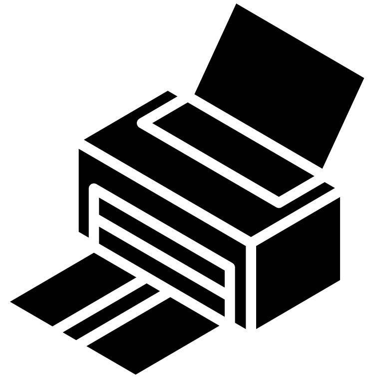
The last step then is to write a summary of the paper in a document that contains all your summaries. It's okay to include your opinions, but you'll want to reference this later so ***try*** to keep it coherent.

EXAMPLE PAPER SUMMARY EXAMPLE PAPER SUMMARY

Title: How to read an academic paper - a non definitive guide
Authors: Andrew McNutt
Venue: n/a
Venue Year: n/a
Date read: February 16, 2021
Summary:
In this zine the author tries to summarize how to read a paper well, but in doing so becomes unable to remember how to read a paper well and makes a bunch of jokes wrapped in "okay" (if a little pretentious) graphic design.

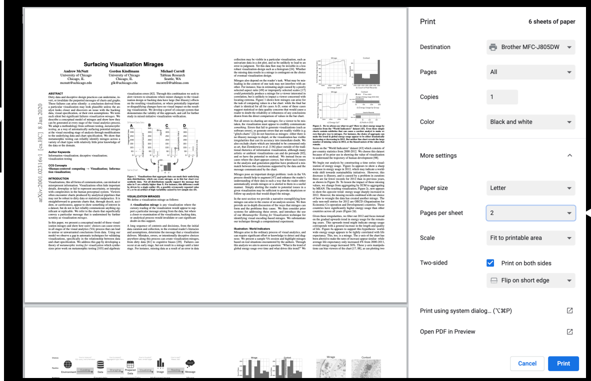
EXAMPLE PAPER SUMMARY EXAMPLE PAPER SUMMARY

Don't write too much, this is just for you. You're not turning it in and can't get points



STEP 1. PRINT THE PAPER

Getting the paper in print is valuable as it transforms the document from something ephemeral into something tangible and definite.



Chrome (and lots of other programs) lets you print two document pages per print page!

MAGIC!

In addition to looking cool this also gives more space to write notes in the margins

STEP 2.

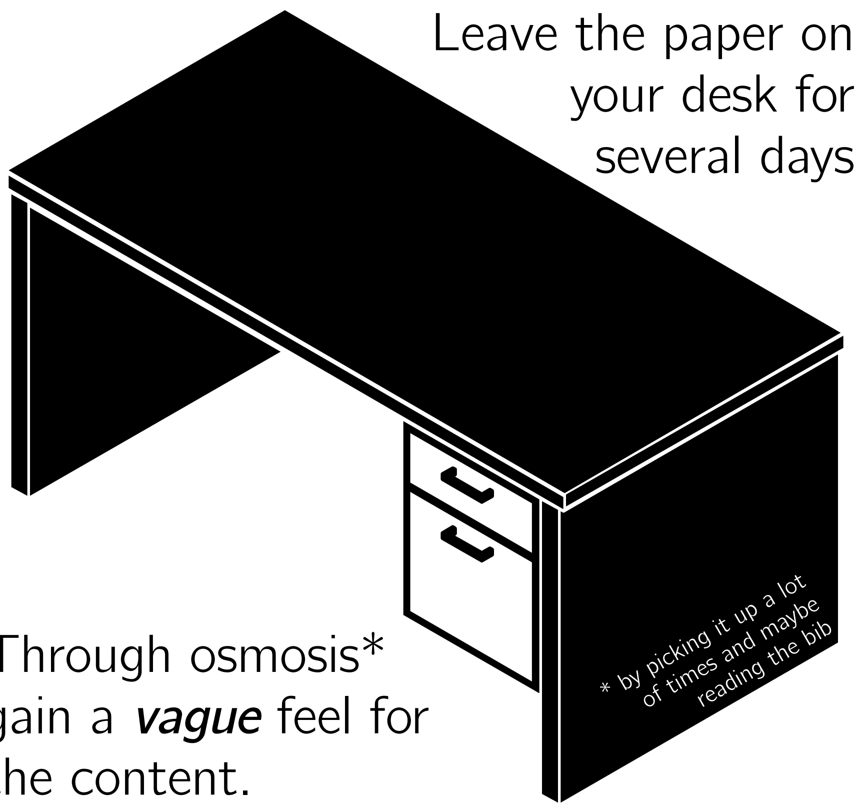
PRE-READ

(PROCRASTINATE)

It's okay if your attention wanders, just stop reading briefly and take a lil time for yourself.

Look out the window, go to the bathroom, look at twitter, do what you need. Reading is hard!

Leave the paper on your desk for several days



Through osmosis* gain a ***vague*** feel for the content.

TAKE LITTLE BREAKS

You can take a day or two break if you really need, but be careful, this can be a way to just not read the paper. (Or to forget what's actually going on)

(This step can be skipped if you are excited)

FIGURES CAN BE

OBJECTS OF GREAT DEPTH

OR

WORTHLESS TRASH.

EQUATIONS ARE MUCH THE SAME, (DEPENDING ON THE PAPER AND PAPER TYPE)

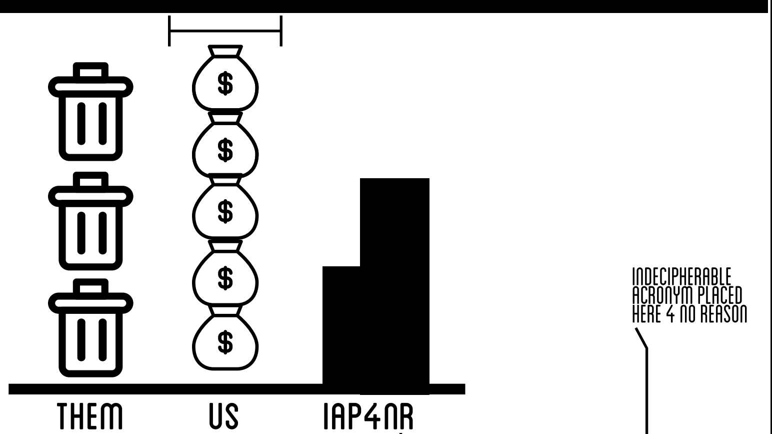


Figure 7: Our analysis indicates that the stilted setup constructed to demonstrate our system is great worked out awesome!

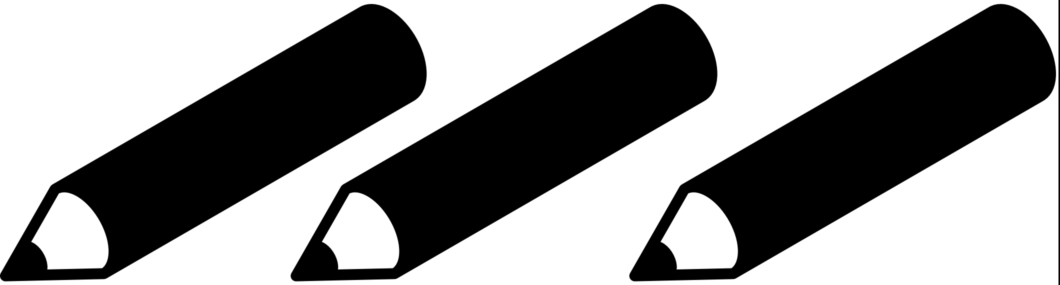
You can often assess which of these they are by looking at how carefully they are made*. Spend time with the good ones, forget the others.

*my highschool physics teacher was fond of saying *good work looks good.*

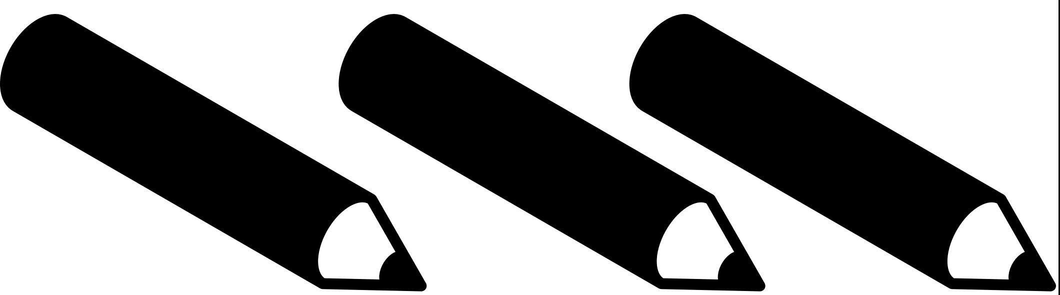
STEP 3.

READ**THE**PAPER
EAD**THE**PAPERR
AD**THE**PAPERRE
D**THE**PAPERREA
THEPAPERREAD
HEPAPERREAD**T**
EPAPERREAD**TH**
PAPERREAD**THE**
APERREAD**THE**P
PERREAD**THE**PA
ERREAD**THE**PAP
RREAD**THE**PAPE
READ**THE**PAPER

STEP 3A.



GET YOURSELF A PEN



Red and green are great because they are easy to see. (Highlighters are not allowed, because you can't write notes)

GATHER MATERIALS

The learnability of the system and mixed-modality remained a primary difficulty. Users typically struggled to figure out how to bridge the gap between text and GUI at first, however, by the end of the session, all users were competent in both regimes. For instance, most subjects iteratively refined their solutions to the box plot task, modifying both code and GUI values to address developing hypotheses. P3 noted that the system required a non-trivial level of computational and visualization literacy. Fortunately, users could seemingly bootstrap their knowledge to overcome these hurdles. P4 noted that the integration between the *"code body and the point-and-clickable GUI is really tight and also good for reinforcing learning"* and that *"If you know how to do something in one form, you can do it and watch how it changes the other side of the tool."*

The scope of this study was small. We merely sought to demonstrate that real users of similar systems could approach the mixed-modality UI found in Ivy. While these results suggest that this combination is promising, further investigation is required to understand its utility in the context of more developed interfaces. Once our system reaches maturity, we intend to conduct a study comparing it with standard analytics tools, such as Tableau or Excel.

6 DISCUSSION

In this paper, we described how *parameterized declarative templates*—a typed abstraction layer over JSON specifications—can serve as a basis for a multimodal UI to create and explore visualizations. Ivy-style templates may help in the organization and reuse (**G4**) of existing visualization corpora (per Sec. 5.1). Vega and Vega-Lite have garnered ample popularity, and new declarative visualization grammars are being actively developed [42, 49, 96]. As the availability and use of these grammars continues to proliferate, there is opportunity for shared platforms and tooling between languages, which we explore in our grammar-agnostic templates.

The integration of features in our prototype appears to be accessible to users with modest experience in both visual analytics systems and Vega/Vega-Lite (per Sec. 5.2). Users were able to make effective use of affordances for exploration found in our shelf building UI and fan out (**G2**), and were able to utilize the capability of templates to improve the ease of use (**G1**) and reuse (**G4**) of declarative chart specifications while maintaining their flexibility (**G3**).

We believe that this multimodal approach has value for a variety of use cases. Exposing a connection between GUI and programmatic API may enable analysts to self-serve their chart creation needs. If a particular chart form is not available (but is constructible by one of the supported grammars) then they can create it for themselves, rather than requiring reliance on engineering resources. This connection between text and GUI appears to help users learn and comprehend JSON-based charting grammars, which may be unfamiliar or difficult to understand. The repeatable customization found in templates might also, for example, enable practitioners (e.g. data journalists) to explore designs in a structured manner that does not violate their organization's visual identity.

6.1 Limitations and Future Work

The version of multimodal visual analytics found in our prototype has its share of limitations. The strength of each modality in Ivy is only as good as its implementation, which can render artificial barriers between what users expect and what is supported (e.g. P3 expected a pivot table). And while Ivy encompasses chart choosing, shelf building, and textual specification, it does so at the cost of an increased learning curve. However, we believe that this difficulty is not endemic to multimodal systems, and that through attentive design the experience of using the system can be made easier.

In making our template-based approach more viable for practical use, it is easy to imagine a variety of system improvements—such as additional template parameter types (e.g. color schemes or inline data fields), drag-and-drop interactions for refactoring and abstracting specifications (e.g. [29, 47]), as well as enriching the ways in which changes made in one modality are reflected and explained in the others. Beyond these, we highlight below several avenues for future research.

6.1.1 Language Extensibility. Ivy is designed to be extensible: support for each specification language is defined through a standardized interface, which includes a JSON Schema describing the syntax, a JavaScript rendering function for the language, and rewrite rules to help users abstract specifications. Our implementati[illegible]tly supports a small set of languages (Vega, Vega-Lite, [illegible]om[1], a[illegible] a simple table language), which, in future work, we w[illegible] to increase so as to support a greater variety of tasks.

Our grammar-agnostic template framework provides a standard set of abstraction and data manipulation mechanisms—which may reduce the need for grammar designers to define their own—and novel UI features for exploring candidate templates (catalog search) and encodings (fan outs)—which may facilitate more efficient and consistent exploration. Our system, furthermore, hoists the burden of data transformation out of the rendering grammar (albeit with a currently-limited set of transforms), which would otherwise require each grammar to implement its own data manipulation logic. As users move between templates (specified in possibly different grammars), their settings (including filters) are mapped from one template to the next via role and order-based heuristics. Future work could enable translation between supported grammars, which could yield opportunities for education and portability.

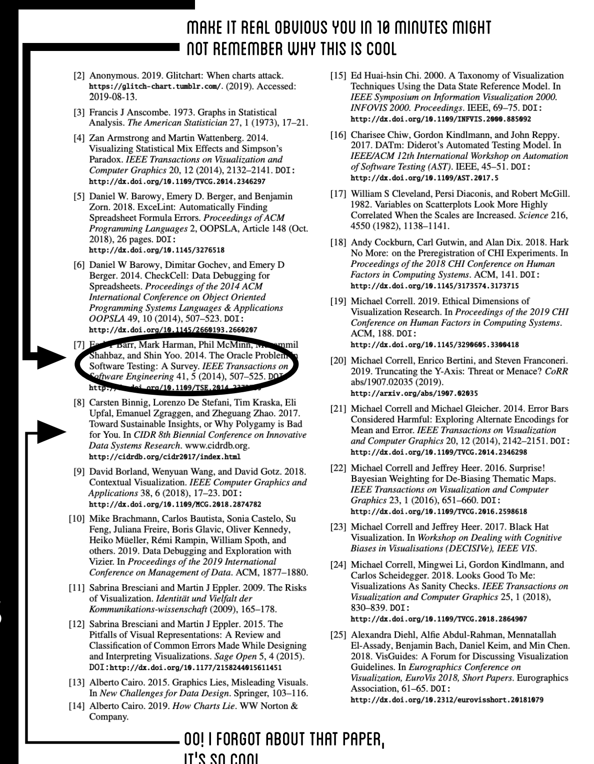
Despite the benefits of language-independent functionality, there are also benefits to taking domain-specific knowledge into account. Language-specific rewrite rules—part of the extension interface, described above—are one such example. Language-specific knowledge could further be used for recommendation, as well as data manipulation and presentation concerns. For instance, when a data field is dragged to a drop zone in Lyra [74] the appropriate type of scale is automatically inferred [75]. Lyra is able to offer this functionality because it has a model of the grammar being manipulated, a functionality which our approach currently lacks.

A lack of context and content-aware automated guidance is a key limitation of our design. Yet, it should be possible to identify a richer extensibility API, while still allowing each language to benefit from the abstraction and UI concerns shared by all. Such an API would enable us to combine domain-specific chart recommendation (Sec. 2.2.3) with Ivy's domain-independent type-based exploration (Sec. 4.4), as well as embrace new interaction modalities.

[1]Whi[illegible]mplementing support for Atom we extracted the language in the original [illegible]tion into a standalone library: https://www.npmjs.com/package/unit-vis

Anytime there are footnotes circle them in the text, that way you can be sure you didn't miss them.

Anytime there are cites you like or are curious about, go to the back and circle them. If you are motivated, look them up later, nbd if not. Reading bibs is its own reward



[2] Anonymous. 2019. Glitchart: When charts attack. https://glitch-chart.tumblr.com/. (2019). Accessed: 2019-08-13.

[3] Francis J Anscombe. 1973. Graphs in Statistical Analysis. *The American Statistician* 27, 1 (1973), 17–21.

[4] Zan Armstrong and Martin Wattenberg. 2014. Visualizing Statistical Mix Effects and Simpson's Paradox. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2132–2141. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346297

[5] Daniel W. Barowy, Emery D. Berger, and Benjamin Zorn. 2018. ExceLint: Automatically Finding Spreadsheet Formula Errors. *Proceedings of ACM Programming Languages* 2, OOPSLA, Article 148 (Oct. 2018), 26 pages. DOI: http://dx.doi.org/10.1145/3276518

[6] Daniel W Barowy, Dimitar Gochev, and Emery D Berger. 2014. CheckCell: Data Debugging for Spreadsheets. *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications OOPSLA* 49, 10 (2014), 507–523. DOI: http://dx.doi.org/10.1145/2660193.2660207

[7] Ea[illegible] Barr, Mark Harman, Phil McMinn, M[illegible]mmil Shahbaz, and Shin Yoo. 2014. The Oracle Problem [illegible] Software Testing: A Survey. *IEEE Transactions on Software Engineering* 41, 5 (2014), 507–525. DOI: http:/[illegible]doi.org/10.1109/TSE.2014.[illegible]

[8] Carsten Binnig, Lorenzo De Stefani, Tim Kraska, Eli Upfal, Emanuel Zgraggen, and Zheguang Zhao. 2017. Toward Sustainable Insights, or Why Polygamy is Bad for You. In *CIDR 8th Biennial Conference on Innovative Data Systems Research*. www.cidrdb.org. http://cidrdb.org/cidr2017/index.html

[9] David Borland, Wenyuan Wang, and David Gotz. 2018. Contextual Visualization. *IEEE Computer Graphics and Applications* 38, 6 (2018), 17–23. DOI: http://dx.doi.org/10.1109/MCG.2018.2874782

[10] Mike Brachmann, Carlos Bautista, Sonia Castelo, Su Feng, Juliana Freire, Boris Glavic, Oliver Kennedy, Heiko Müeller, Rémi Rampin, William Spoth, and others. 2019. Data Debugging and Exploration with Vizier. In *Proceedings of the 2019 International Conference on Management of Data*. ACM, 1877–1880.

[11] Sabrina Bresciani and Martin J Eppler. 2009. The Risks of Visualization. *Identität und Vielfalt der Kommunikations-wissenschaft* (2009), 165–178.

[12] Sabrina Bresciani and Martin J Eppler. 2015. The Pitfalls of Visual Representations: A Review and Classification of Common Errors Made While Designing and Interpreting Visualizations. *Sage Open* 5, 4 (2015). DOI:http://dx.doi.org/10.1177/2158244015611451

[13] Alberto Cairo. 2015. Graphics Lies, Misleading Visuals. In *New Challenges for Data Design*. Springer, 103–116.

[14] Alberto Cairo. 2019. *How Charts Lie*. WW Norton & Company.

[15] Ed Huai-hsin Chi. 2000. A Taxonomy of Visualization Techniques Using the Data State Reference Model. In *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings*. IEEE, 69–75. DOI: http://dx.doi.org/10.1109/INFVIS.2000.885092

[16] Charisee Chiw, Gordon Kindlmann, and John Reppy. 2017. DATm: Diderot's Automated Testing Model. In *IEEE/ACM 12th International Workshop on Automation of Software Testing (AST)*. IEEE, 45–51. DOI: http://dx.doi.org/10.1109/AST.2017.5

[17] William S Cleveland, Persi Diaconis, and Robert McGill. 1982. Variables on Scatterplots Look More Highly Correlated When the Scales are Increased. *Science* 216, 4550 (1982), 1138–1141.

[18] Andy Cockburn, Carl Gutwin, and Alan Dix. 2018. Hark No More: on the Preregistration of CHI Experiments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 141. DOI: http://dx.doi.org/10.1145/3173574.3173715

[19] Michael Correll. 2019. Ethical Dimensions of Visualization Research. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 188. DOI: http://dx.doi.org/10.1145/3290605.3300418

[20] Michael Correll, Enrico Bertini, and Steven Franconeri. 2019. Truncating the Y-Axis: Threat or Menace? *CoRR* abs/1907.02035 (2019). http://arxiv.org/abs/1907.02035

[21] Michael Correll and Michael Gleicher. 2014. Error Bars Considered Harmful: Exploring Alternate Encodings for Mean and Error. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2142–2151. DOI: http://dx.doi.org/10.1109/TVCG.2014.2346298

[22] Michael Correll and Jeffrey Heer. 2016. Surprise! Bayesian Weighting for De-Biasing Thematic Maps. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2016), 651–660. DOI: http://dx.doi.org/10.1109/TVCG.2016.2598618

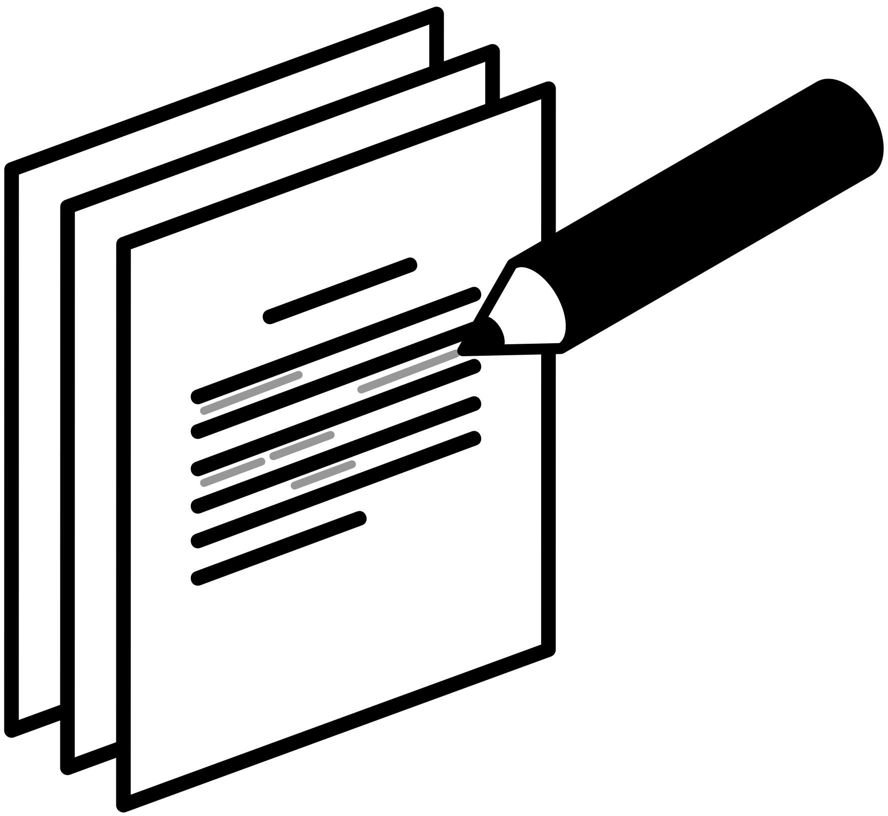
[23] Michael Correll and Jeffrey Heer. 2017. Black Hat Visualization. In *Workshop on Dealing with Cognitive Biases in Visualisations (DECISIVe), IEEE VIS*.

[24] Michael Correll, Mingwei Li, Gordon Kindlmann, and Carlos Scheidegger. 2018. Looks Good To Me: Visualizations As Sanity Checks. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2018), 830–839. DOI: http://dx.doi.org/10.1109/TVCG.2018.2864907

[25] Alexandra Diehl, Alfie Abdul-Rahman, Mennatallah El-Assady, Benjamin Bach, Daniel Keim, and Min Chen. 2018. VisGuides: A Forum for Discussing Visualization Guidelines. In *Eurographics Conference on Visualization, EuroVis 2018, Short Papers*. Eurographics Association, 61–65. DOI: http://dx.doi.org/10.2312/eurovisshort.20181079

UNDERLINE STUFF AS YOU GO

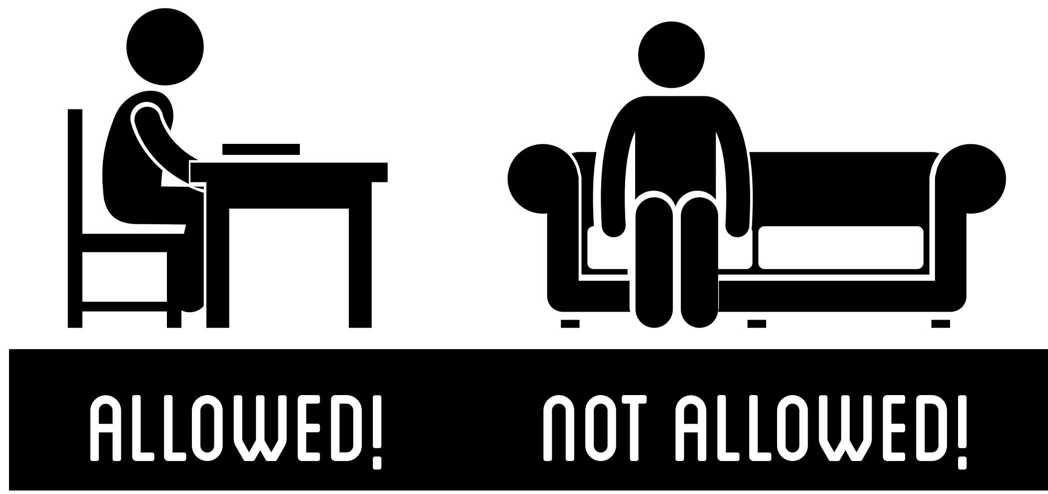
NOT EVERYTHING, BUT ANYTIME ANYTHING SEEMS TO BE SAID WITH **'RHETORICAL FORCE'**



This can yield a lot of underlining. This is unlikely to be useful later, but it is ***useful now***, as it helps you not gloss over long or boring sentences.

STEP 3B. FIND A LOCATION

Sit down at a table with the paper.



Try to remove all distractions: put away your laptop, you can have music (but try to keep it non-distracting*)

Posture is super important! Sitting on a couch isn't great for comprehension

* drones** are great for this

** Such as the works of Pauline Oliveros

STEP 3C.

ACTUALLY

READ THE FUCKING

PAPER

It takes awhile! Be patient!

(like 1-4 hrs)