Data Cleaning IS Analysis, Not Grunt Work

Also, most data cleaning articles suck


Randy Au
Sep 15

Hi new newsletter readers! This is a completely atypical epic-length post. It's only when the muses come with a bucket of inspiration, which is quite rare.

TL;DR: Cleaning data is considered by some people [citation needed] to be menial work that's somehow "beneath" the sexy "real" data science work. I call BS.

The act of cleaning data imposes values/judgments/interpretations upon data intended to allow downstream analysis algorithms to function and give results. That's exactly the same as doing data analysis. In fact, "cleaning" is just a spectrum of reusable data transformations on the path towards doing a full data analysis.

Once we accept that framework, the steps we need to take to clean data flow more naturally. We want to allow our analysis code to run, control useless variance, eliminate bias, and document for others to use, all in service to the array of potential analyses we want to run in the future.

It's Thursday already (short weeks are the worst) and I once again asked people on Twitter if anyone had ideas for stuff I should be writing about. One response I got back was: "Cleaning". Seriously, that was it.


Matt Guttman @RealtimeAI
@Randy_Au Cleaning.

September 10th 2020

1 Like

At first I thought that was asking a bit much. Data cleaning is an age-old topic, what could I write in a newsletter that would contribute anything at all to the conversation?

Then I started looking at existing articles about data cleaning… and it got ***annoying***. There's a mountain of stuff all trying to sound authoritative and saying the same things. But as a practitioner, I can see it's not useful. It's like the infuriating "is left as an exercise to the reader" in textbooks. Last time I got this annoyed at a tech writing topic, it spawned a monster SQL optimization post.

Then, I started seriously thinking about data cleaning as a concept, and fell down an entire epistemological rabbit hole.

So let's get started.

Existing Data Cleaning writing is pretty useless

First let's start with stating the problem with existing writing on "Data Cleaning".

Wikipedia's post on data cleaning does a decent summary of the big important qualities of data quality: Validity, Accuracy, Completeness, Consistency, Uniformity. It's also got a section on "process" that's really dry and academic (in a negative way) and won't help you clean any data at all.

Next I'm just gonna sample posts from the top links on Google when I search "Data cleaning". I'll provide links as reference so you know what I'm griping about.

This highly PageRanked one is like a friendlier expansion of the Wikipedia page at the start. Luckily it redeems itself in the process section by listing a big list of example techniques to use to clean data, things like cleaning spaces, dropping irrelevant values, etc. Has some examples and illustrations!

This one is some random blog-like post for some data product site I've never heard of. It's short and mostly "data quality is important, you need to monitor it on an ongoing basis". They're not wrong, but also not that helpful. Then they give a process loop for cleaning, verifying, and monitoring your data… I assume this is because they're selling you some of this functionality in the product.

Next, Tableau gets in on the repetition game, saying why clean data is important, listing out a similar checklist of steps, and tacked on the end are those qualities of data quality! Not bad, but remarkably similar to the other entries thus far.

This one on a data science education site. Says things like "Good data > Fancier algorithms", also cartoon pictures of robots (I like cute things), so that's a plus.

Then it's right off to the checklists, remove bad observations, fix errors, fill in missing values, etc. I sorta get their "selling" skills here, but still.

Pattern-spotting brain senses pattern!

> If you want a highly ranked Data Cleaning post just take the concepts of "Data quality is important", "Audit data, find errors, fix them, monitor, repeat", and "here's the qualities of good data quality". Toss in a few examples. Do it in < 1,000 words.
>
> Profit! (Literally, through ads and product sales)

My biggest gripe is that these are all shallow checklists of "Go find the bad stuff and clean it! Use these tools/techniques. Easy!" If it were so easy, we wouldn't be spending so much time doing it. You're often handed a list of "Good data has these qualities" so just go look at your data and just make sure it has them. If not, find a way to impose those qualities upon the data. Meanwhile every reader is thinking, "so, how?"

Side Note: There's some better discussions if you search for "Data cleaning theory".

There's also efforts to automate data cleaning (usually with the help of AI because obviously AI makes everything better). Part of these efforts stems towards there being just too darned much data and humans can't possibly analyze all of it. But the these products/functions are also aimed at reducing the drudgery of cleaning data without even looking at it. Of which I'm highly skeptical. Automation is important, but so is actually doing the cleaning.

Cleaning is difficult and nuanced work, why do we treat it like laundry? We don't give checklists for how to analyze data, do we? (Oh no, sometimes people apparently do…)

Data Cleaning IS Analysis

I never thought hard about data cleaning until this post. But as I looked at it critically, it's became obvious to me that we've put an unpleasant label on top of a tedious, frustrating part of the data analysis process (who wants to do CLEANING, might as well call it "scooping the data kitty litter"). Then through layers of abstraction and poor education, we've lost track of how critical it is.

We then tell horror stories and have "concerning" research that 80%, 60%, 40%, whatever-percent of an expensive data scientist's time is spent on cleaning data. The stat itself seems more a vague expression of direction than hard truth. Leigh Dodds wrote a more detailed look at that sketchy statistic here.

Whatever the actual truth of the claim is, the implication and call to action is clear — if we made the cleaning process easier, faster, *automated*, we'd hit Data Science Productivity NIRVANA. CEOs can save on headcount and salary. Every employee could do their own simple analyses from a slick GUI. Data scientists would be working on the sexy problems, use the *bestest* algorithms. There'd be overflowing wine goblets and insights will effortlessly rain from the heavens.

Silliness aside, pause and think about what data cleaning actually is. Not the mechanistic definition of "getting rid of errors to increase data quality". Nor the end-goal justification of "having better data quality gives better results". Let's talk ontology.

We perform data cleaning because we suspect there's useful signal about some topic we care about. For example, I suspect the volume of my tweets that complain goes up when an important deadlines looms. But my signal of interest is buried in plenty of noise and errors: my shitposting volume isn't correlated w/ deadlines, my time zone changed over history. I downloaded the data in weird chunks so there might be holes. I write in two languages. I tweet a lot so I can't read all the tweets and hand-code them.

We're doing cleaning because we want to extract the useful signal from the noise, and we decide certain bits of noise "correctable" at the data point level for that purpose. Therefore, we clean those parts so that they don't affect our analysis.

Wait, but that's data analysis!

We just call it data cleaning because somehow it's "not the real analysis I'm doing", it's "the stuff that comes before". The Real Work™ is using algorithms with names, not "find and replace". I'm just forced to do this menial labor step so that I can achieve my true glory as the bringer of truth.

This cleaning work is perceived to be mechanistic, and "not hard". A high-schooler or RA could do this, and they cost factions of MY salary. But considering how complex a 16 year old human brain is, I don't feel that way.

Here's a very typical cleaning situation. *Side note, The embed is ridiculously long so I'm screenshotting the original tweet here. Literally feels like scrolling through Philadelphia for miles.*

The act of cleaning data is the act of preferentially transforming data so that your chosen analysis algorithm produces interpretable results. That is also the act of data analysis.

If our algorithms were so magical that we can just throw in raw text logs in and real insights falls out, guess what, we'd never clean. Our computers don't care how ugly our data is — it's all 1s and 0s. Regression will gladly spit out coefficients to whatever table of numbers you feed it. The real reason for all this cleaning work is the signal-to-noise ratio in raw data is too poor for purpose we intend. We need to improve data quality to amp the signal for our tools to find.

We do things like: decide that every user who misspelled Philadelphia still counts, even if they can't spell their home town. We don't simply use ALL the model features because only 3 matter. We need to exclude users with profile XYZ because we gave them a special discount and they'd bias our A/B experiment on the sales page. All these users from France don't count because we didn't run our ad there. Duplicate events from bug#12345 need to be excluded or it'll bias our results.

All of the above are instances where we've cleaned our data by making a decision. You might classify some as "cleaning" and some as "analysis", but they're all very similar classes of transformations. What gets labeled as "cleaning" vs "analysis" seems more a judgement of its value — the good stuff that makes us sound smart is called "analysis".

But cleaning is mechanical. Analysis has intent!

Analysis implies that we're searching for meaning. Wouldn't cleaning, a step's largely for "fixing quality issues" like typos and collection errors, have no analytical intent and this is the differentiating factor between the two?

Nope!

Because cleaning operations themselves impose value judgments upon the data. Going back to the Philadelphia example, it may seem obvious to you that all those misspellings should be grouped together and cleaned to be the same, correct value. As a frequent analyst of crappy data from open text boxes on the internet, that's my automatic reaction upon seeing it. This is because the work I do generally wants all data from a given location to be grouped together.

But what if you're doing a linguistic study? Those typos are actually extremely valuable to you. Normalizing that variation away would make any analysis on that front impossible. If we didn't store the original data (which is recommended best practice that only some people follow), it would forever be lost.

Choosing to transform the data in any one way necessarily implies you've already chosen to do certain things with it, to the exclusion of others. It doesn't matter if the changes are big (imputing missing values, removing entries) or small (removing trailing whitespace, normalizing date formats).

As an example, one of the largest data generation/collection systems on the planet, the Large Hadron Collider generates so much data it's impossible to store in raw form. So physicists spent untold amounts of thinking and energy, drawing upon decades of domain-specific knowledge, to create custom algorithms to filter and preselect what to even record. It's built to purpose —they literally designed and control the entire toolchain down to the hardware.

> *Some credit is due here: Reaffirming that nothing I ever write is new under the sun, halfway through writing this article, I tweeted about data cleaning and it went viral. Someone in the amazing Twitter data cluster pointed me to this publication from by Katie Rawson and Trevor Muñoz (2019).*
>
> *It's from the data humanities fields (a domain I didn't even know existed despite doing some computational philosophy/sociology 15 years ago) that discusses very similar observations about data cleaning much more cogently. Very recommended read!*
>
> *Also another great person linked me to this paper that compares two broad methods for cleaning data, imposing order upon the disordered. It also goes into how the the fields of economics and*

biology work to preserve the raw data, anticipating that things might need to be reclassified (re-cleaned) in the future.

And yet another awesome person pointed me to danah boyd's work and this paper that discusses a lot of the critical issues with Big Data, including data cleaning. Pointing out that data cleaning is seen as trivial when it has enormous effects on what you can even say with a set of data. Also well worth a read.

Cleaning your data allows you to know your data

I firmly believe that knowing your data is critically important as a data scientist. It's the one thing I tell everyone who asks me for advice. You can only make confident statements using your data when you know what's been collected, what **hasn't** been collected, how it's collected, where are the gotchas, where are the hidden dependencies, etc.

The only way to get that level of familiarity is to do the hard work and get deeply familiar with the data set at hand. People usually attribute this work to be part of the "Exploratory Data Analysis" phase of analysis, but EDA often happens when the data has already been partially cleaned. Truly raw, unclean data often doesn't even function properly within software, because something unexpected usually trips things up. Fixing those issues bring a huge amount of knowledge (and questions) about the data set. (Okay, astute readers would notice that this isn't the only way, the best way to get ultimate familiarity with a data set is to actually **collect** the data yourself, but that's a topic for another day.)

Most of us don't work in rigorous institutions that carefully document every single detail about data collection like say, the US Census Bureau. Manuals that can answer minute questions about methodology, data quality, and cleanliness don't exist in our world. Also, let's be honest, how many of us who use the Census data actually read any of those documents? Definitely not me.

I'm going to put forth the claim that cleaning the data is a prerequisite to getting a rock solid understanding of the underlying data. If you delegate that responsibility someone else, whether it's another human or a machine, you put yourself at risk of doing something dangerous with your data. The pragmatic question is how much risk we are willing to assume, and whether you're even aware that you're undertaking risk.

In many cases, the implications of that danger is small — few people normally care if you misinterpret the census data to make a bad map to tweet out. For expedience, you might be willing to trust that the Census did a good job maintaining high quality data and just accept that you might do something stupid because you didn't read a footnote somewhere.

By taking the data quality to be good enough based entirely on trust, you can save time and skip ahead to EDA. That's a pretty tempting position to take when using a well-known, reliable source for simple uses. But even with a trusted source, inconsistencies do exist and can be found later. Hopefully it doesn't upset your findings.

But if the map you created was to be used for making decisions involving life and death? Or it's for serious academic research in search of Truth? Then the stakes are much higher and you need to be absolutely sure your findings aren't some data artifact. Don't trust anyone else's cleaning.

But we can't do cleaning for every analysis! That'd take forever! It won't SCALE!

Not for every data analysis. For every data **SET**. The knowledge learned from cleaning doesn't disappear overnight. Code you wrote to execute the cleaning isn't lost.

Cleaning is a powerful method of becoming familiar with a data set, and also generating the needed data set to arrive at interesting results. If you know the data so well you can navigate it blindfolded, you won't be doing manual cleaning anyways. By then, you've got code and tools and guides already built, and you're deeply familiar with the tooling.

I haven't been saying we can't automate the cleaning operations we choose to do. The problem with automation is that it can divorce understanding the underlying data from the analysis and interpretation of the data. When I query a data warehouse, I'm not familiar with the multitude of pipelines written by unknown engineers that each do very specific transformations on terabytes of data per hour. That always puts me in a very uncomfortable position of not being 100% confident what my work rests on.

But if you're doing the cleaning and writing the cleaning code and use it to automate, that's not a problem. You're the one who found all the issues and implemented all the fixes. Hopefully you **document and communicate** that knowledge so that others may benefit and save the time and effort. Otherwise you're just leaving others in the position of having to trust your analytical judgement.

No one is safe from change

Out of all this work, the one thing everyone needs to worry about is needing to continuously keep an eye out for situations where the data quality shifts from under our feet. This is an ongoing risk for any long-running data operation.

It doesn't matter if it's an automated or manual process, if the underlying data collection process changes, then all bets are off. So expect to occasionally have to revisit cleaning operations.

It doesn't help safeguarding against these situations is extremely difficult in a modern distributed-systems infrastructure setup. Something is always in the process of changing or breaking and these have unpredictable side effects.

Let's Redefine Data Cleaning as "Building Reusable Transformations"

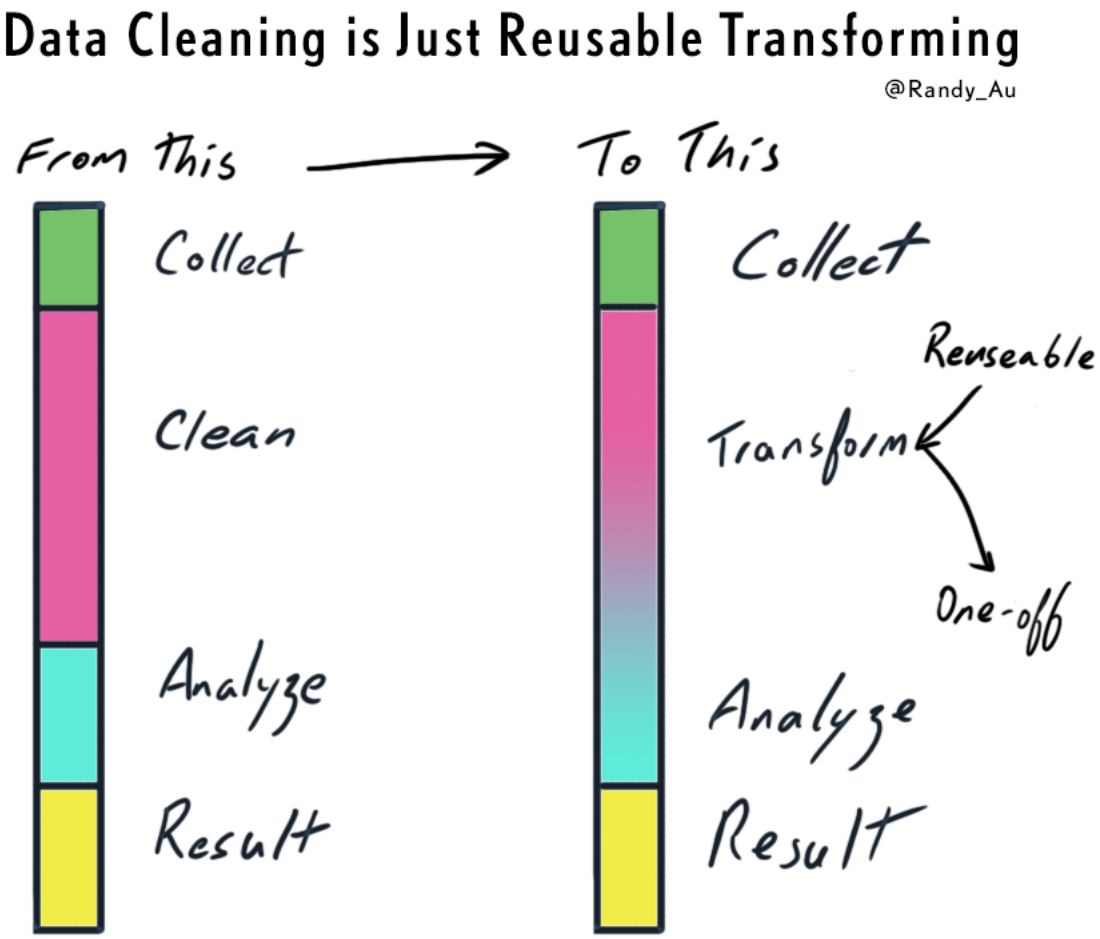
Once you elevate the status of the data cleaning work to be equivalent to "full data analysis" where it should have been all along, we're left in this unsettling position where everything is analysis. The compulsion to break the megalith analysis process into phases will be too strong to resist. We already have Exploratory Data Analysis as a phase, something must fill the void.

I propose that we start thinking of data cleaning as "building reusable transformations", for two reasons:

First, analysis is using transformations to pull insight out of data. We shuffle things around a certain way (e.g. clustering, SVM, decision trees), squint at the data through a certain lens (e.g. regression, statistical inference), and we see previously unknown patterns and truth.

Second, typical data cleaning operations are done very early in the analysis process in preparation for doing "Real Analysis". The only reason we can do this is because those early cleaning transformations are reusable across a broad array of analyses. That reusability decreases on a sliding scale as we home in closer to the final analysis.

So the act of "cleaning data" is actually building a library of transformations that are largely reusable across multiple analyses on the same data set. At some largely arbitrary cut-off point, you hand off the data to other people to use.



Okay, now that I've ranted a bunch about how data cleaning is a critical part of the analysis process, let's work through the process of doing data-cleaning-as-analysis.

Doing Data Cleaning "Better"

When you start thinking about data cleaning as "the steps I need to take to allow my analysis algorithms to produce useful results" instead of "here's a list of attributes and bugs I need to check for before I can use my data" the world looks a bit different. We can work backwards from our end analysis goal.

First off, never do permanent changes unless you're REALLY sure

When we embark on cleaning and analysis, we're going to be making a bunch of decisions early on that we might regret in the far future. We want to avoid that. So the first rule is to always keep a copy of the original data around if at all possible.

I've actually zipped up and archived separately a raw dataset before starting work just so I couldn't accidentally trash it.

It's only when there are overriding constraints (like the LHC's volume issue) should you consider throwing data away permanently. You better be damn sure about the quality of your data cleaning process before doing so. Data storage is relatively cheap compared to the regret of not being able to undo any mistakes and needing to collect new data.

Next, always leave an unbroken analytical paper trail

I once wrote about this topic in more detail before, but you always want a paper trail of your analysis. Every step in the transformation from raw data to final presentation should have links/references to prior steps. This serves to remind you of what you did to complete an analysis in order to reproduce it at a later date, and it also allows you to answer questions about how the analysis was done ("did you account for X?") months, years after the fact.

In the age of Jupyter notebooks, colab, and version control, there's not much excuse not to have the trail available. Over time, your library of cleaning transformations will become robust enough that you can pack it up into a library.

With these basics in mind, we can embark upon cleaning/analyzing our data.

Goal 1: Fix things that will make your analysis algorithm choke

We all should be very familiar with this sequence of events: you get your data formatted, put it into your software of choice, hit the "Analyze the things!" button, annnnnnd…. it crashes — invalid input, unexpected null, negative time durations, incorrect array size. BUGS.

This class of issues at first blush seem the most mechanistic and easily dealt with via automation. These are the random NULLs that crop up, the ridiculous emoji that slipped through the form validation, the duplicated IDs that should have been unique, data entries that have timestamps that imply time travel because logging system clocks weren't in sync. Usually these are easy to find because your data pipelines crash or give otherwise nonsensical results.

When people speak derisively about data cleaning, they're thinking about these issues. There's countless variations and they're all annoying to deal with. Usually you fix one, get hopeful that the analysis you've spent days on will finally show a result… and then a new one pops up.

The fixes to many of these issues often seems easy too. Just strip out the Unicode, turn all NULLS into blanks, flat out ignore the duplicates, just please let my code run!

Sadly, in a production environment, we can't paper over all these issues. Those quirks are bugs. Sometimes they're innocent and you can safely strip out the Unicode. But sometimes they're the result of a more insidious bug, one that introduces bias into the entire system. Like perhaps for some unfathomable reason, ONLY names with an emoji wound up in the dataset due to a second logging bug.

The problem is that there's no way to know where a bug falls on the innocent<->disaster spectrum just from looking at the data. You have to put in the legwork to analyze the issue and figure it out yourself. This'll slow your work down considerably, it might even force you to throw your data set away! It's often thankless and undesirable work. It's why we hate doing it.

But if you fail to do it, your analysis potentially has no legs to stand on.

Goal 2: Reduce Unwanted Variation

In the perfect ideal world of platonic spherical cows falling in vacuums, the only variance that would exist in a dataset are the "differences that matter". Everyone in group A gets an exact score of 35 on the test, everyone in group B gets exactly 95. Group A and B are clones and differ only by treatment T. Boom, we have perfect experimental results.

Obviously, we must deal with variance in the real world. But you can imagine ways to "clean up" data that would give maximal signal to various algorithms like regression, k-means, etc., while still staying within ethical and statistical boundaries. Simple examples are the fixing of spelling and normalizing punctuation. More advanced cases include clustering values into categories.

When people are doing "Analysis" with the big A, they're often able to find ways to control this sort of variance using methods with names like "feature selection", "normalization", or "feature generation", all of which involve trading some variance information for an algorithmic benefit down the road. When they don't feel like boasting about it, they'll shove it in a footnote/appendix and call it "cleaning".

Some of the measured variance is useful to us, and some can be removed without affecting our goal analysis. Knowing the difference is an important part of the craft of doing data work.

Goal 3: Eliminate Bias (where you can)

Bias is our enemy. We want our work to be robust and generalizable, and biases in the data ruin that. If you're going to budget the amount of time and brain cycles you spend on doing data preparation, spend it the most of it here.

Many common operations used to control bias include throwing certain data entries out, filling in missing values, or correcting/resampling/remapping values, all in the name of controlling peculiar data biases that creep into the raw data.

This is a place where **subject matter expertise is critical**. Experts on a topic are better able to identify places where bias potentially creeps into a data set. These people know where to spot the overt biases in the collected data: data collection error (duplicates, etc), internal tester accounts, population oversampling, self-selection in respondents, etc. But experts also should know how to identify the implicit structural biases of the data set, what *hasn't* been collected and can come up with strategies to minimize the damage from those issues.

Domain knowledge will also tell you whether something needs to be cleaned or not for your task at hand. If we didn't care what city people were signing up for loans from, we'd never have to worry about cleaning over 50 different ways to misspell "Philadelphia". If we suspected we might need it, we'd put some energy into fixing it.

With all the current talk about ethical use of machine learning and algorithmic bias, this aspect of cleaning data is more critical than ever. Bad things can happen if biased data gets into ML systems, like computer vision systems still have trouble with handling dark skin colors.

By the point you reach this level, many of the cleaning operations you're working on should start to get quite specific to your intended end goal and it's hard to tell if you're cleaning or doing analysis. The steps are getting progressively more involved and need more rigorous justification.

I'm making a data set for others, how do I know what to clean?

Unless you know what people are going to use the data for, you won't know. Even if you know, you'd still must guess at what is likely to help their analysis and not hurt it. By far the most important thing to do is to **fully document** every cleaning decision made in case it's important in the future.

Knowing where to stop and hand off to the next person is the hardest part of the process. You have to peer into the future and anticipate what people are going to do. Since I can't even predict what analysis I'd do myself, let alone others, you won't get this 100% correct and will have to make adjustments. It's best to plan for that iteration up front.

Thanks!

Hopefully this mega-post helps tweak your way of looking at cleaning data a little. I admit that I didn't even stop to think seriously about "What is cleaning?" until this post and it prompted all sorts of very interesting questions. There're plenty of smarter people who have thought and wrote about this better, like those papers I linked earlier.

About this newsletter

I'm Randy Au, currently a quantitative UX researcher, former data analyst, and general-purpose data and tech nerd. The Counting Stuff newsletter is a weekly data/tech blog about the less-than-sexy aspects about data science, UX research and tech. With occasional excursions into other fun topics.

Comments and questions are always welcome. Tweet me. Always feel free to share these free newsletter posts with others.

All photos/drawings used are taken/created by Randy unless otherwise noted.