DataCamp Writing Guidelines

You may be asking yourself, "why on earth am I reading a document about *writing*, I know how to write!" And you may be right. However, DataCamp is a diverse group; there are many ways to "write right," and it's unlikely we are all on the same page. It is worthwhile from a product standpoint to ensure that language is consistent when a user flows from a promotion to a course to a community post and so on. The following is a set of suggested guidelines to follow for any piece of writing that is accessible by the public. Some are relevant to most/all departments, and some are specific to certain teams.

Basic Writing Guidelines

Acronyms

Acronyms are **everywhere** in data science (DS). However, just because an acronym exists, doesn't mean it's appropriate to use. So, when should you use an acronym and when shouldn't you? Within a document, you should **only** use an acronym after it's introduced and then you should *exclusively* use that acronym from there onward. What does this mean? This means that from this point on in *this* document, I should never use the term "data science" again, but exclusively use DS.

Here are other acronym rules:

1. Unless it's a *very* commonly used acronym (e.g., MRI, FBI, SCUBA, etc.), acronyms should *never* be used in titles, headers, etc.
2. Figure captions are considered isolated text; you must **always** redefine acronyms within a figure caption. If you have 12 figures within the same document and the same acronym is used in all 12 captions, you have to define that acronym 12 times. Why? Figures (and their captions) should speak for themselves; your reader should **not** be required to skim through your text *or even look at* more than one or your figures to understand **any** of your figures.
3. Outside of situations where acronyms are necessary for succinctness, such as in scientific documents where several similar nouns may be used, making up your own acronym is generally not a good idea.

There are likely more exceptions that I don't know about, please feel free to add them.

American versus British English

DataCamp uses American English (AE). While you may be aware that Brits believe that *o* and *u* make the perfect couple and have a terrible fear of the letter *z*, you may be completely unaware of other more subtle differences between British English (BE) and AE.

The Oxford comma: Also known as the serial comma, the Oxford comma is probably the most notorious example of differences in punctuation between the two camps. Despite its ridiculously Anglican name, the Oxford comma is *not* preferred by users of BE but *is* a mainstay of AE. It's so important to Americans that the internet is littered with GIFs about it. Here is an infographic I found that does a much better explaining the Oxford comma than I ever could:

- U.S. Government Printing Office Style Manual

Dislike, avoid and omit:

- The Associated Press
  (most news outlets follow the AP Stylebook)
- The Economist
- New York Times

Why use it?

- Helps avoid ambiguity
- Matches the natural speech pattern of pausing before the last item in a series
- Makes lists easier to comprehend

Why omit it?

- Sometimes doesn't help clarity and can introduce ambiguity
- Redundant—the preceding conjunction serves the same purpose as the comma
- Takes up more space

IRONY ALERT!

The PR department at **Oxford University no longer uses** the Oxford comma (although Oxford University Press still does).

Commas can be cool!

The band
VAMPIRE WEEKEND
has a song called "Oxford Comma."

The Oxford comma has a Facebook page with over
29,000 likes.

So what should YOU do?

Spelling differences: The obvious ones include the aforementioned *o* versus *ou* and *z* versus *s*, but there are few other differences to note. Here is a fairly comprehensive list of spelling differences between AE and BE.

Slang and Idioms: This is unlikely to be a huge issue, *but* it's best to be aware that lots of the things we say are **not** universal. Furthermore, this is not just a BE versus AE issue; it's regional/cultural/personal/etc., and it's best to err on the side of clarity. For example, I have a friend from Sydney who likes to say "I'll take a sticky beak" to mean "I'll have a look at it." I don't think any other human says this. It's one of the weirdest phrases I've ever heard a person utter.

Latin & Greek

Latin is something that we use more often than we think, our instructors use *a lot*, and many people *do not* use it properly. If you're not 100% clear your usage is appropriate, opt for English.

Latin Phrases/Abbreviations
e.g. = *exempli gratia* = for example
i.e. = *id est* = that is = in other words
et al. = *et alii* = and others
etc. = *et cetera* = and so on

Also, there are slight punctuation differences when using Latin phrases in AE and BE. I don't think these are a big deal personally, but as usual, just be consistent.
AE: e.g., **BE:** e.g.
AE: i.e., **BE:** i.e.

Sentence Case versus Title Case

How you structure your titles may have never occurred to you before, but it's worth being aware of the two title styles: sentence case and title case. You are probably more familiar with title case, where all "important" words are capitalized. Books, stories, movie titles, etc., almost exclusively use title case. Sentence case, on the other hand, is where only the first word and proper nouns are capitalized, but all other words are in lower case. Scientists generally prefer sentence case because the *vast* majority of scientific publications require sentence case for both titles and headers. The most important thing here is to be consistent. The American Psychological Association (APA) has guidelines on creating titles in either case.

Numbers

Since we work in data, it follows that we have to write out numbers *a lot*. There are a few guidelines for writing numbers that we should adhere to because they enhance clarity when writing. Here is a very well-written article on How to Write Numbers by the aptly-named Grammar Girl.

Grammar, Syntax, and Punctuation

I'm not going to go over this. My knowledge isn't so vast that I can adequately cover this and furthermore, the topic is so huge that no one would ever read this. Needless to say, grammar, syntax, and punctuation are all important for clarity and professionalism. I'll leave you with this:

Data Science Jargon/Conventions

Many words and phrases are very specific to DS and engineering, and these are littered throughout our courses, tutorials, marketing material, job listings, etc. While there may be no *correct* way to express these, we as a company should commit to consistency. The following list is by no means exhaustive, so **please** feel free to add terms that you believe belong here. Furthermore, the (first) spelling listed here is *suggested*, if you have strong feelings that something should be written another way, feel free to **challenge the current spelling with supporting evidence**. However, if you think it's "data set" and not "dataset," be ready for a fight.

DataCamp (**not** *Datacamp, Data Camp, or Data camp*)
Backend, back end, back-end
Frontend, front end, front-end
Fullstack, full stack, full-stack
dataset, data set, data-set
dataframe, data frame, DataFrame
database, data base, data-base
Data Science, data science
scatterplot, scatter plot
dotplot, dot plot
real-world real world
x-axis, X-axis, x axis (with hyphen so that x and axis always render on the same line/time)
Principal component analysis (*not* principle)
KNN, knn K-NN k-NN, k-nn
Preprocessing, pre-processing

....

Technology Specific Jargon

There are a few special cases where consistency among DataCampers may come into direct opposition with the technologies we teach. In these scenarios, best practice deems we follow suit with the leading authority for those languages, i.e., RStudio and Anaconda for R and Python, respectively. Below are the technologies that we currently have on Campus with a list of the "proper" way to write some DS terminology. Bear in mind that these nuances apply to the beginning of a sentence as well within a sentence, e.g., "pandas is great!" is correct, but "Pandas is great" is incorrect. This is because they are proprietary terms; one is the correct word, and the other is not.

Please add any and all jargon here. I don't know the nuances for all theses languages, so all help is welcome.

R

data frame

Python

DataFrame
NumPy
pandas

SQL

...

Git

...

Shell

...

Scripts and Subtitles/Closed-captions

The most important aspect of a script is how well it conveys the content. We're all looking out for this, and I think we're doing a damn fine job. However, subtitles *are* how a large portion of our students are getting our content (*are we tracking this?*) and we need to ensure that we do not consider subtitles a "secondary asset," but a primary feature of our product. Who else is taking their subtitles seriously?

The BBC and Netflix for starters. Fortunately, we have been making a strong push to ensure our courses do not violate the Americans with Disabilities Act. But remember that subtitles ensure access not only for the Deaf and Hard of Hearing community but for many non-native English speakers.

Standards

As it turns out, there are really no standards for the writing element of subtitles, and best practices largely deal with elements of subtitles that are out of our control, such as the number of characters that appear on each line and where line breaks occur. If we could find out more about how our subtitles are rendered, we could check whether our subtitles follow these practices, and if not, what we can do to change them?

Rendering Issues

As the Content Team knows, the current subtitles are limited. Transcribing coding terms is awkward, and the current approach is very literal, e.g., `pd.DataFrame` turns into "pd dot data frame", and `group_by` turns into "group underscore by". We need best practices for our scripts so that students don't go from one course to another and end up confused by the different ways we tackle this problem. Other rendering issues/questions include:

- Numbers: Periods cause line breaks, so how do we deal with decimals?
- Functions: Same issue as with decimals.
- Should we include () to indicate it's a function?
- Only some instructors actually use the words "underscore" or "dot" when describing functions/methods/attributes that use them. Should we only write those words when they *say* them or should we be writing them when they *mean* them?
- It would probably make sense to find out how to get an entire function/method/attribute call to render on the same line, otherwise potentially only half the "code" will actually be visible, which isn't ideal.

Colloquialisms: Writing for reading and writing for speaking are two different beasts and our instructors take full advantage of this. This means that they will often use "filler" words in place of a pause and will emphasize words by USING CAPITAL LETTERS. We need an approach that either includes these nuances or shuts them down. Here are a few things I've come across frequently:

- OK, Ok, Okay
- So...
- Ellipses in general
- Listing off bullet points (which of course don't render properly due to the aforementioned decimal issue)

Multiple Instructors: This scenario is one of the few things out there where there is a standard practice, which, fortunately, will be very easy to implement. Current standards say that a speaker should be "introduced" the first time they speak in a given "scene", but subsequent lines do not require the speaker's name, just an indication that the speaker has changed.

But what does this look like? Let's look at an example from the concluding video for Supervised Learning with scikit-learn where Hugo and Andy Müller have a back-and-forth exchange. The first words spoken by each speaker are indicated with their name in brackets. Subsequent lines by each speaker show the text preceded by hyphens, which indicate that there is a back-and-forth exchange between the two speakers.

> *[Hugo]: ...You have sure come a long way!*
> *[Andy]: To recap, you've learned the fundamentals of using machine learning techniques...*

> *-- You have learnt the concepts of underfitting, overfitting, and...*
> *-- You've also gained first-hand experience at using regularization...*