Designing For Data Scientists

A User Experience Designer's Notes on Designing Data Science Tools

Robert Coyle | April 2017 | www.cremeglobal.com

Table Of Contents

Introduction

Starting out as a User Experience Designer in a new domain is tough. You need to quickly learn everything about your user - what they do and why they do it. In a technical domain like Data Science this can be especially challenging.

This document tries to structure my notes over a two year period into an overview of issues around designing tools for Data Scientists. Also consider that I am coming from the perspective of a consultancy business that designs predictive models and accompanying software.

The document is aimed at other UX designers or Product Managers starting out in a "design for data science" type role.

If you have any questions, suggestions for content or spot any errors you can get me at: robert.coyle@cremeglobal.com or my LinkedIn.

Our Company: www.cremeglobal.com
Our Product: Expert Models

Data Science the Profession

The following chapter is an extended persona aimed at introducing you to the profession of Data Scientist.

Types of Data Scientist

Doing user research for design in data science is greatly helped by the fact that Data Scientists tend to turn their statistical powers on themselves. Because of this you can get hold of papers like “Analysing the Analysers” (Haris, Murphy and Vaisman, 2012, O’Reilly). This paper surveys 250 practitioners and in doing so proposes four personas (archetypes) of the Data Scientist. This is a great starting point for a Designer working on their personas. If you find from your own research that you would add or split a persona, go ahead. That’s what personas are for.

There are four personas proposed by the paper.

1. **Data Business Person**
2. **Data Creative**
3. **Data Developer**
4. **Data Researcher**

Lucy the Data Business Person

Summary: Lucy works as an analytics manager and spends much of her time in meetings rather than on hands-on statistics.
Other traits:

- Enjoys working on data and visualisations when there is time.
- Has an engineering background and masters in business.
- Sees part of her role as translating difficult data science lingo for the company.
- She has a good business head.
- She is up on the latest developments in Data Science.

Monika the Data Creative

Summary: Monika works as a Data Scientist for a major consultancy and has a reputation as the go to person for creative solutions and what is trending.

Other traits:

- Attends PyCon and hackathons.
- Has her own meetup group on deep learning.
- Believes in open data.
- Studied computer science before mastering in statistics.
- Blogs and tweets regularly and has a big following.

Karolina the Data Developer

Summary: Karolina is a programmer and systems person at heart. She likes solving difficult problems.

Other traits:

- Writes clean machine learning code.
- Knowledge of multiple programming languages.
- Open source software contributor.
- Particularly interested in database technologies.

Shane the Data Researcher

Summary: After spending many years in academia while calling himself a statistician Shane has entered the workplace as a Data Scientist for a marketing firm.

Other traits:

- Good head for research and doing thorough work.
- Experience presenting at conferences.
- Builds predictive models for his company based on sales data and social data.
- Wants to demonstrate how statistics can change his business.

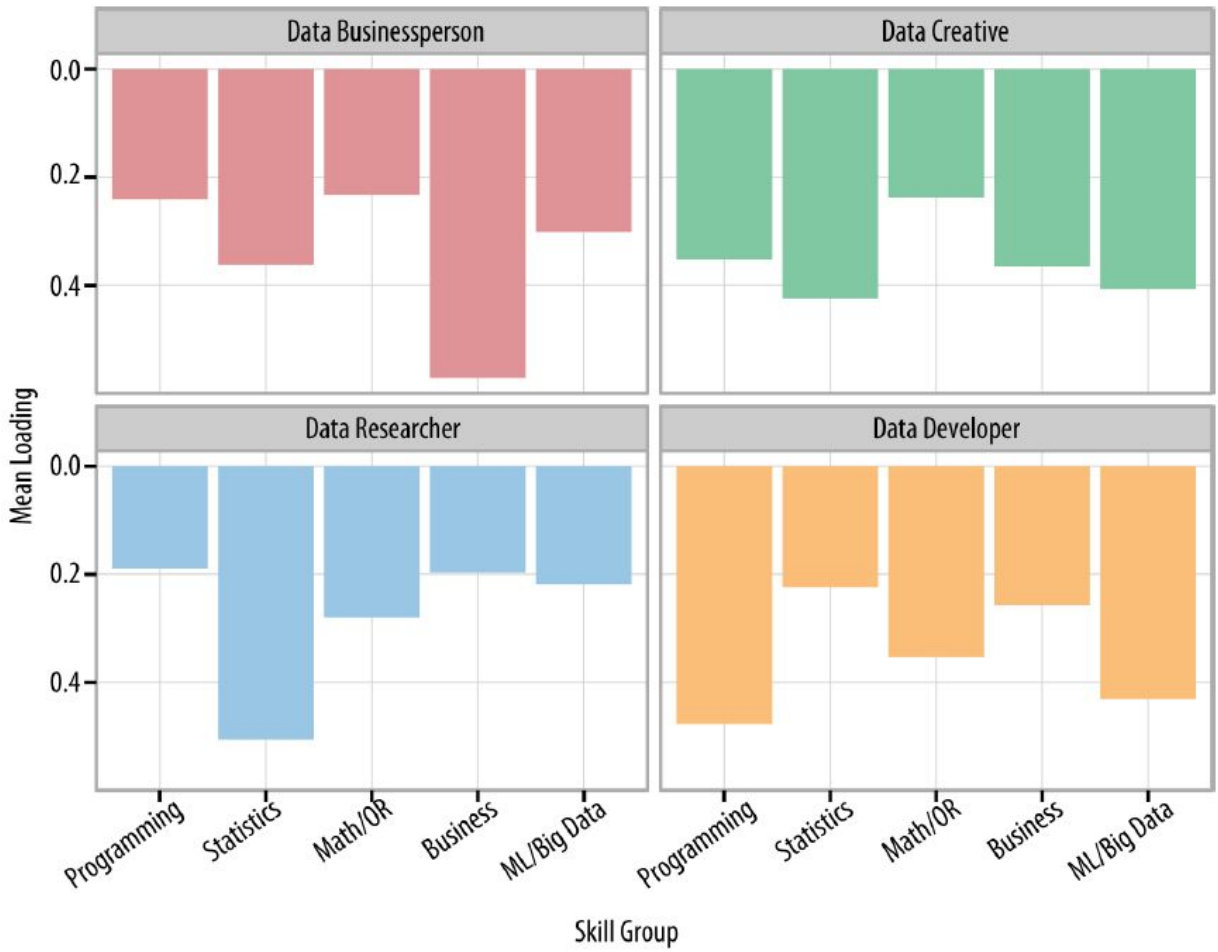


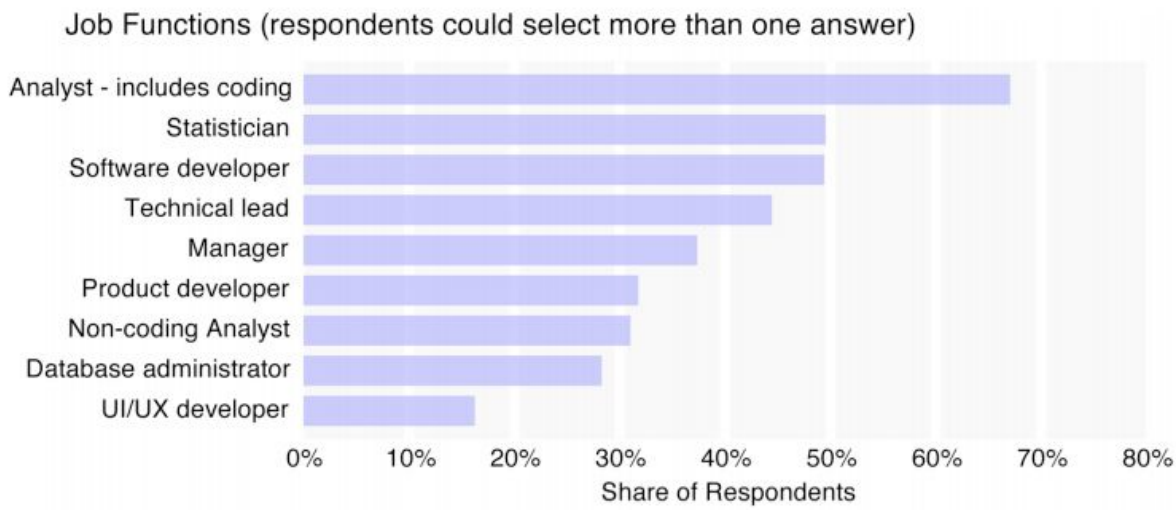
Chart taken from Analysing the Analysers (Haris, Murphy and Vaisman, 2012, O'Reilly)

The Data Scientist's academic background

As a Designer it is good to know the educational background of your users. O'Reilly's "Analysing the Analysers (2012)" provides the following information which might now be a little dated. Data Scientists often come from a science background rather than a tool based education. 40% of undergraduate degrees and 70% of masters degrees studied sciences. Scientific backgrounds like physics or astronomy teach good statistical thinking. On the other hand social / economic /political scientists make good Data Scientists too. They have a better ability to aggregate data, understand causality and present findings.

The varied job functions in data science

O'Reilly conducted a survey in 2014 of workers in data science, analytics and management with 816 respondents in 53 countries. 40% were from tech companies. 40% came from companies with over 2500 employees. Here are the results.



What this says when thinking about design for Data Scientists is that some people who identify as Data Scientist are not technical individual contributors. If you are designing tools for teams of Data Scientist, it is good to be aware of this diversity as it will influence the uptake of your designs.

Data Scientists as Programmers

A Data Scientist may spend a percentage of their time programming. As a Designer we must understand this element of their workflow. We may eventually design programming tools for Data Scientists.

What is the difference between a Data Scientist who programs and a traditional programmer? This is an illuminating comparison.

	Data Scientist	**Regular Programmer**
Languages Used	Mostly R & Python	All Languages
Where will the program	Often a model's results	Mostly production

be used?	ends up in a report. Some is for production software	software
Size of programs	A model might be only 100 lines of code in one file	Often 1000's of lines of code in multiple files
Programming patterns used to organise code. Best practices.	Reproducibility	Various patterns including OOP and MVC
Debugging	Basic use of stop point and commenting out	Advanced tools for tracking logs and examining dependencies
Performance of program	Not a consideration	Yes, a consideration
Use of libraries	Maths libraries like NumPy	Wide and varied libraries
Manipulate data with code	When yes, at a very advanced level.	When yes, at a basic level

The takeaway is that the Data Scientist is mostly using programming to write their models, using maths libraries and getting statistical output. They would not be considered software engineers and to push code into production they would often pass their code over to actual software engineers to be optimised or rewritten and incorporated into an application.

Collaboration with other Data Scientists on programming

From my research I found that Data Scientist as Programmers do not collaborate with each other in the same way traditional Programmers do. In a traditional programming team, work on a piece of functionality might pass from person to person over time. For Data Scientists they seem to have more exclusive ownership over their code and might only collaborate with a colleague when choosing which statistical function to use. We can suppose that this lack of collaboration is down to the deep level of meaning behind that code that is not obvious. A model script may be quite short but it's meaning or written specification might take up 10's of page to describe.

Data Scientists do tend to review each others results to sanity check them. A collaborator might spot an unusually low number in results or an unusual looking chart output.

Finally, a Designer should be aware of the new breed of Data Scientist that gets involved with highly collaborative communities like Kaggle and share and collaborate more freely .

Coding environments preferred by Data Scientists

Data Scientists write their models in relatively sophisticated coding environments. A simple text editor or software like Sublime or Coda are not typical, unless for optimising or web development. A good example of a Data Scientist coding environment is MATLAB. Why? MATLAB has the following features:

- Access to easy to install packages that Data Scientists rely on.
- The user interface is tailored to Data Scientists. The layout in MATLAB is four panels used for 1) Code 2) Output. 3) Data input (list of variables). Also variables of output are created. 4) Browsing folders. Other tools like R-Studio are almost identical in layout. These panels can be laid out in various ways.
- Graphs are linked with the data. Clicking areas in the graph filters the data.

Newer Trends: Notebooks

You can think of a notebook in data science as a report with live code, charts and data embedded. There are a number of notebook applications available, such as Jupyter.

The idea of a notebook is very appealing:

- They are a quick way to explore data once set up
- They bring the code and its deeper meaning closer together
- They accelerate development by allowing chunks of code to be run/tested in isolation
- Notebook reports are superior to static documents
- They can be centers of knowledge for a data science team
- Companies like Quantopian use notebooks as an interface / documentation to their dataset.

While notebooks have been around forever they are exploding in use these days. If you would like to know more this article is a great primer. As a Designer you might consider

how notebooks inform your design. What do they imply about the needs of a Data Scientist?

Use of Libraries

It is worthwhile considering how Data Scientists use statistical libraries versus how traditional developers think about libraries. This knowledge has affected my design decisions so I feel it's worth sharing.

- Data Scientists writing models tend to change libraries in search of a function that works (fits their data). Not all LogNormal function implementations are the same it turns out. The Data Scientist would test with different libraries. In R there are seven methods for calculating weighted percentile. There is a default but the modeller might want to pick any of the other six.
- Libraries are often badly documented and so need to be tested. An alternative to testing is to look at peer reviews
- Use of libraries can result in legal cases if they are copyrighted.
- A Data Scientist might rely on 5-6 good libraries to get their work done.
- Data Scientists are not as aware of version numbers of libraries in the way that developers are. The maths is stable in their view.
- Libraries they tend to save development time and help with standardisation.

A long model run is a job

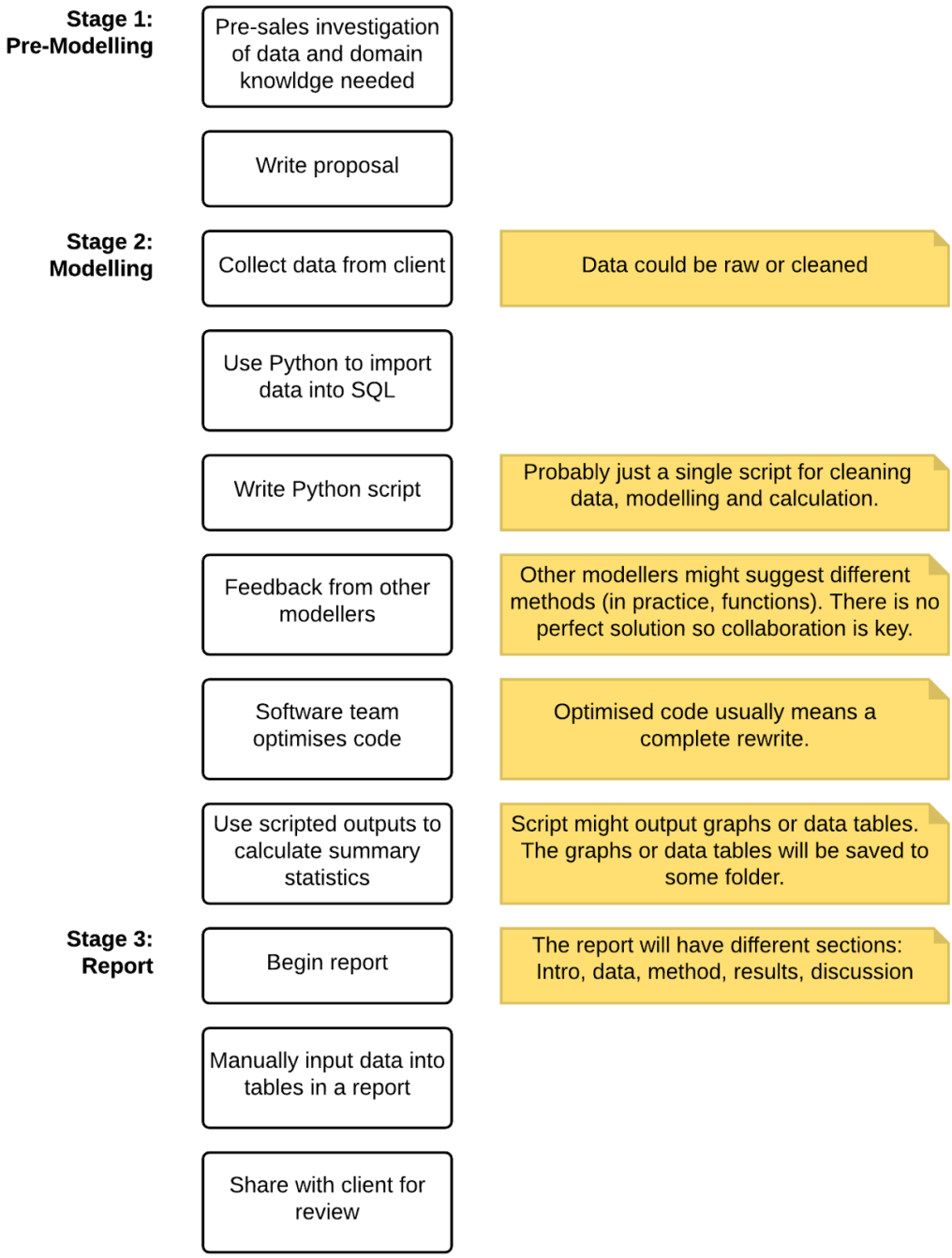
When a Data Scientist runs a model they may not get results on the screen right away. Models can take seconds or hours to run. A Designer might consider this when creating interfaces around models. Other design considerations:

- If output takes a long time to compute consider the concept of job queues or progress indicators. Input, computations and libraries can make a model run take longer to complete.
- Models can potentially figure out how long they will take to run based on intelligent learning.
- When developing a model a Data Scientist is not so interested in saving results. Consider a debug mode for running that does not save results.
- Bad libraries or code might increase run time.
- Modularisation may help - if there are several stages of analysis, perhaps only the later ones need to be repeated.

A “consultancy” based Data Scientist’s workflow

As a Designer you should try to understand the workflow of the Data Scientist you are designing for. Not all workflows are the same. Below I have put together a process flow for an actual Data Scientist working for a consultancy with some of the boring bits taken out. On the following page I make a number of other observations.

Consultancy based Data Scientist’s workflow



Other observations

- The consultancy Data Scientists interviewed reported spending their time on: 2-10% model development, 40% data cleaning, 50% scientific report writing and business development. The actual time spend on modelling was surprising low. Note that most people who self-identify as a Data Scientist do not write models or program - although that is changing.
- The ability to communicate well and understand the business is essential for the Data Scientist. A consultancy Data Scientist will often deal directly with a client.
- Clients are often other Data Scientist or domain specialist in medium to large businesses. Data Science is often a service sought by large entities where even small optimisation to their business gives worthwhile returns.
- Often the output of a data science project is a report. Any modelling and programming is solely aimed at getting a statistical result. In 2017 very few models make their way into production software.
- The report, as just mentioned is often the culmination of their work. It is an important artifact. As a Designer you might consider:
    - How does your design relate to their report? Do you as Designer help create it or are some of your assets included in the report somehow.
    - Data Scientists look to plots and graphs more than tables of data, but both are often included in results.

Data Scientists in Machine learning are different

Machine learning is a fastly growing area in data science. The explosion of interest in artificial intelligence these days is mainly about machine learning. The output of a Data Scientist's work in this area is often destined to go into production software. Brian O'Mullane of Creme Global in this quote describes the difference very well between the consultancy Data Scientist and this new kind:

"For me Data Scientists have a remit to understand the business and produce insights / reports. They build models that don't need to run fast / may not be used by customer. Machine learning teams have an engineering bias. Their results are put into production for the customer to use. They train models that run in real time."

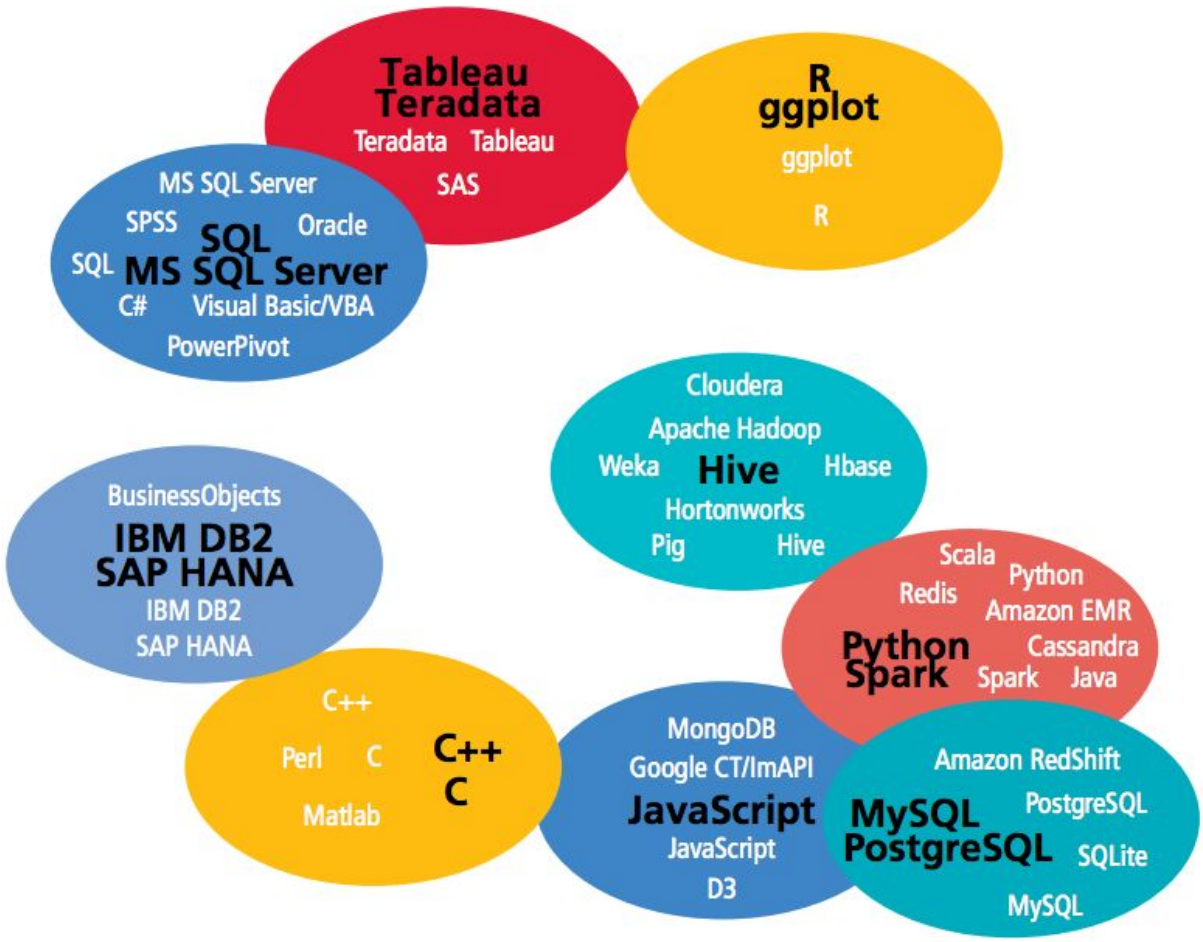
The workflow around writing a model

For Designers writing tools for Data Scientists it is useful to zoom in on the actual steps a Data Scientist takes in writing a model. Their use of visualizations for exploration is particularly interesting. These steps are of course a gross simplification:

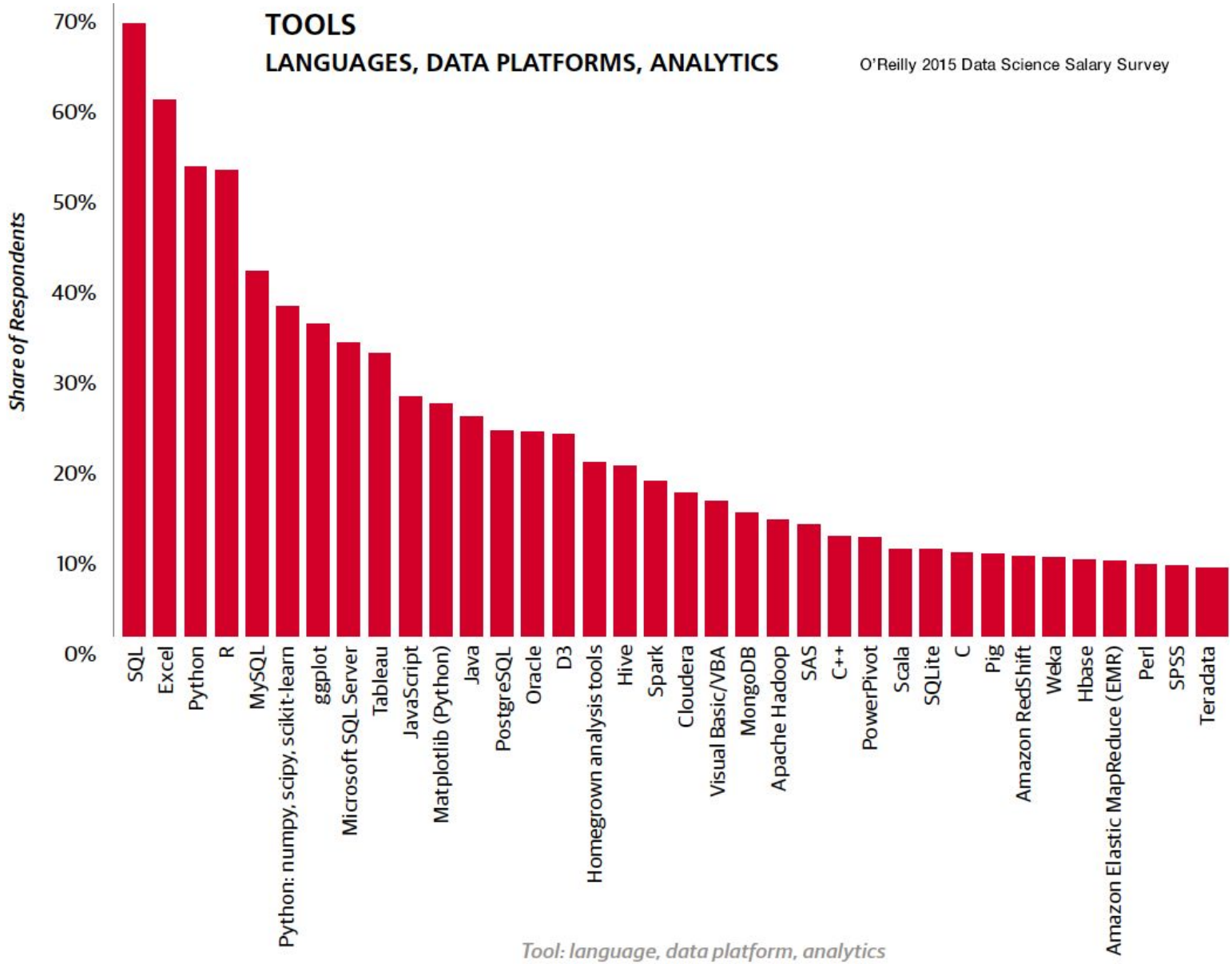
1. Starting with data. Explore with visualisations and statistics.
2. Write model. You might start by building a model with all variables then remove variables if they do not play a role. The fewer variables there are the better a model will be. This improves robustness. A clean robust model can then be applied to other datasets.
3. Create final visualisation and statistics if this model is destined for a report.

Tools used by Data Scientists

The [O'Reilly 2015 Data Science Salary Survey] give great insight into tools used by Data Scientist. This cart shows tools clustered by use. Correlations within clusters are higher than between cluster.



This chart shows the % usage of tools by respondents to O'Reilly's survey of 600 practitioners (2015)


TOOLS
LANGUAGES, DATA PLATFORMS, ANALYTICS
O'Reilly 2015 Data Science Salary Survey
Share of Respondents
70%
60%
50%
40%
30%
20%
10%
0%
SQL
Excel
Python
R
MySQL
Python: numpy, scipy, scikit-learn
ggplot
Microsoft SQL Server
Tableau
JavaScript
Matplotlib (Python)
Java
PostgreSQL
Oracle
D3
Homegrown analysis tools
Hive
Spark
Cloudera
Visual Basic/VBA
MongoDB
Apache Hadoop
SAS
C++
PowerPivot
Scala
SQLite
C
Pig
Amazon RedShift
Weka
Hbase
Amazon Elastic MapReduce (EMR)
Perl
SPSS
Teradata
Tool: language, data platform, analytics

Tool: language, data platform, analytics

The special role of spreadsheets

Note the special place spreadsheets like Excel hold within a Data Scientist workflow. Often novice Data Scientists will default to spreadsheets without exploring other more appropriate tools. Spreadsheets are used in two different ways by Data Scientists.

1. As a database, if scale and complexity are within limits. Beyond a certain limit the Data Scientists would use a formal database, usually some implementation of SQL.
2. As a statistical analysis tool and for basic modelling. We look further at this scenario in the next section.

A Designer might consider designing features that work with spreadsheets rather than against them, or at least acknowledge the reason for Excel's ubiquity and perhaps consider it an education challenge if they wish a user to switch to their tool.

Working with Data

We have already covered that a Data Scientist can spend on average 40% of their time wrangling data and are better at working with data than regular programmers. Their main tools for manipulating data are Excel and SQL. Following are a couple of topics that are worth being aware of.

The Sharing of data

Whether it be companies, bigger projects with multiple stakeholders, or academics, the data they use is mostly proprietary. There may never be a GitHub for data. This culture is caused by the legal challenges around data. Even sound ideas like data obfuscation can be regulated away.

"The major concern is even if you can obfuscate data, you will have a hard time from your legal department," says Alexander Linden, a research director with the Gartner analyst firm.

A Designer might consider the sensitivity of data when building data science tools. For example is sharing the data in question on shaky legal ground?

Open Data

Open data is quite a big topic in Data Science as many solutions rely completely or in some part on open data.

Wikipedia says: *"Open data is the idea that some data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents or other mechanisms of control.'*

As a Designer you might need to upskill in the various conditions around open data. A good place to start is by reading up on Tim Berners Lee's 5-Star-Data initiative. Many open data sources use this ranking. You can also watch Tim at TED talking about this initiative.

Managing Data

Managing data is a massive topic that is outside the scope of this document, however I did want to hint at a couple of sub topics in this area. A great paper in Nature titled **The FAIR Guiding Principles for scientific data management and stewardship** details the problems and solutions around "Supporting discovery through good data management". The paper is aimed at discover of academic data but it's discussion around topics like metadata are universal.

Also, there are entire industries around managing data, such as the field of Master Data Management.

Data Protection Regulations are a big topic. For example in Europe in 2018 we will see the arrival of **The 2018 European Data Protection Regulation -** *'The EU General Data Protection Regulation (GDPR) is the most important change in data privacy regulation in 20 years'*. This Gartner report highlights some of the coming changes.

Data is messy and political so as a Designer it is worth getting to know the issues.

Hierarchical Data vs Flat Data

I wanted to include this topic because as a Designer it caught me unaware a couple of times. Some definitions:

A hierarchical data *model is a data model in which the data is organized into a tree-like structure. The structure allows representing information using parent/child relationships: each parent can have many children but each child only has one parent (also known as a 1:many ratio).*

A flat file database *is a database that stores data in a plain text file. Each line of the text file holds one record, with fields separated by delimiters, such as commas or tabs. While it uses a simple structure, a flat file database cannot contain multiple tables like a relational database can. Source*

So what's the problem? In my experience when a product design is more or less displaying a table(s) of data, if you don't initially understand if the data is hierarchical vs flat you can make the wrong design choice. Data can also come in a flat format but is representing something hierarchical. Often our development team have suggested a better design as they have looked at the data and understand it's structure. I once suggested a spreadsheet type UI for some data without looking at the data. What was really needed, and this was suggested by one our of developers, was a tree picker.

In summary, look at the data and understand whether it is hierarchical or flat.

The spread of statistical thinking

It is worth mentioning to Designers the transformation in companies towards data centric thinking is comparable to the way design thinking is catching on. A much repeated quote that has been around a while captures the sentiment:

"If we have data, let's look at data. If all we have are opinions, let's go with mine."
Jim Barksdale, former CEO of Netscape.

This Gartner report alludes to the spread of data-centric thinking and practices into the general working force.

“Traditional business intelligence (BI) and analytic models (systems) are being disrupted as the balance of power shifts from IT to the rest of the? business, according to Gartner, Inc. The rise of data discovery, access to multi-structured data, data preparation tools and smart capabilities will further democratize access to analytics and stress the need for governance. Gartner predicts that by 2017, most business users and analysts in organizations will have access to self-service tools to prepare data for analysis."

A designer might be aware of the growth of the non-professional statistician in the area of Data Science.

Designing for Data models

What is a model

WikiPedia says this: *"A statistical model is a class of mathematical model, which embodies a set of assumptions concerning the generation of some sample data, and similar data from a larger population. A statistical model represents, often in considerably idealized form, the data-generating process."*

In other words, a model is an ideal representation of a system built on everything we know to be true about the data with the hope that this representation (model) will help us make guesses about other and future behaviour.

What relevance does this definition have for design? For one it implies that the Modeller is interested in a process in the real world. How will the insight from a model be used? How does this affect your design? Imagine software for modelling traffic aimed at making civil engineering decisions. Knowing this is the final use of the model, a Designer might bring insight to the design by looking at the topic of civil planning or traffic in general. Design is about empathy and here is a good place to show it.

To find out more about the how statisticians approach model writing you can see our process at Creme Global here.

The spreadsheet Problem

Spreadsheets are a ubiquitous tool in data science. Spreadsheets are not only used for data manipulation, they are also used for statistical analysis and even running basic models. A team might have a spreadsheet sheet with macros that run an analysis on data. The user can be anyone from a Data Scientist to an analyst working in marketing , finance or any industry that models data. The common activity of forecasting is essentially a type of modelling.

Risks of statistical analysis with spreadsheets

Users stretch the limits of spreadsheets to the point where they are used to perform tasks beyond their ability to perform correctly. IBM carried out useful research on this topic. Their findings are summarised in their unambiguously titled paper The Risks of Using Spreadsheets for Statistical Analysis. Here are some of the insights:

- A typical spreadsheet will have a restriction on the number of records it can handle. If the scale of the job is large, a different tool like an SQL database might be more appropriate.
- If you only need a superficial review of your data, a spreadsheet may be a suitable tool. But if you suspect that there is valuable information in your data that isn't immediately obvious, or if you need to perform a detailed analysis or find hidden patterns, a spreadsheet will not give you the functionality you need.
- Another factor to consider is the degree of accuracy required. Spreadsheet results can be unreliable, especially on large datasets and/or for complex calculations. If absolute accuracy is required, a spreadsheet may not suffice. Instead, a different, more reliably accurate tool should be considered.
- If the task is simply to analyze a limited quantity of historical data, a spreadsheet will do. However if you want to make reliable forecasts or draw trends, especially if they involve large dataset, then there are much better tools.
- Spreadsheets are prone to errors. A number of studies have been made concerning the frequency of errors in spreadsheets. Based on these studies the indication is that 90 percent of all spreadsheets contain at least one error. These include stealth errors that pass the attention of experts and can go undiscovered.

Type of errors

If we look at the type of errors spreadsheet users encounter we see more clearly the advantages of moving to a formal model

Functional Errors: These are errors that break the spreadsheet and give an error message.

Outlier Errors: The spreadsheet still works but an expert spots that these numbers can't be correct.

Stealth Errors: These errors in results even pass the attention of experts. Often in statistics no one know what correct results should look like. Errors like these can go undiscovered.

"IBM studies reveal that 90 percent of all spreadsheets contain at least one error."

Causes of spreadsheet errors

Some of the primary causes include:

Logic: Relating cells incorrectly might mean a logic error. Using the wrong function or operator are logic errors.

Copying formulas: Copying a formula might fall foul of unseen cell relationships. Hand typing a formula is also dangerous.

Copying numbers on top of formulas: Copy a constant number over an invisible formula in a cell and the formula is wiped out along with other cell relationships.

Using the wrong function: Some spreadsheet functions have dangerously similar names.

Leaving out data: Most spreadsheets setups will not give validation error if you have not completed full rows of data and have left some cells out.

Adding incorrect data: Incorrect data entered can often pass unnoticed. Ideally the entry would cause an output effect that is very noticeable - for example accidentally entering a date rather than a number - however this is not always the case.

Excel patches: Tragically patches have caused errors to existing sheets as shown by this report.

Accuracy: Spreadsheet have been shown to have accuracy issues when executing complex maths, even when error free. Excel has gotten criticism for its shortcomings in statistical analysis.

Complexity: Spreadsheet analyses are often made up of chains of calculations. It is very easy to alter links in these chains without realizing and "breaking" the analysis.

"Professional statisticians continue to write books with titles like 'Statistics with Excel,' but they now warn students not to bet their jobs on Excel's accuracy."
Bruce D. McCullough, "The Unreliability of Excel's Statistical Procedures," Foresight, (February 2006) Vol. 3, 44-45.

Other spreadsheet issues

Handling missing data values: A user here might try to enter a zero value which would upset the findings of a median value. If they enter a string it might be misinterpreted by an equation as a zero or something else entirely. One would at least need to come up with a standard for denoting missing values and stick to it. Mark Lambe here at Creme Global pointed out a releated danger: *"Excel's default is to try to handle / interpret ambiguities rather than throw an error. This is a strength for simple stuff but becomes a disadvantage as complexity increases"*

Categorical data with hidden meaning: Imagine you enter the value -1 or -2 in a cell to denote two different types of error. This categorical data needs to be documented somewhere. What if the author of the spreadsheet moves on?

Labour caused by focusing on cells: Spreadsheets are cell centric and modifying a cell can require the changing of formulas or data across a range of cells/rows/columns.

New data and equations: Equations can be set to extend automatically for new data, but sometimes this is not desired. Either way this will be a consideration and possible source of new errors.

Conclusion

Using spreadsheets for serious statistical work can be more trouble than it's worth, and sometimes the user never even finds out about it, but reaps the negative consequences.

Furthermore in Excel the Data Scientist loses the reproducibility coding gives by way of versioning and pull requests.

For Designers working in data science, be aware of the ubiquity and consequences of spreadsheets and be able to bring this knowledge into your design workshops.

Models built with data science tools

Data integrity

If the spreadsheet were compared to a hand saw, then a formal model is like a CNC machine (automated cutting machine) that creates robust reproducible output. That is the kind of accuracy often needed in data science. Following are some of the features that can be designed into formal modelling software.

We talked earlier about how excel can introduce all kinds of problems into data and models. Formal modelling software can protect against some of these problems. As an example, our software Expert Models has the following checks on imported data.

1. Tables are defined by type. This allows us validate against known field / datatype pairs.
2. Checks for null entry errors.
3. Checks for non type errors like incorrect ranges.

Variety of interfaces

If a model grows beyond the confines of a report, and ends up in a web application, an interface can manifest in a few ways. Let us first define a model as a program that models an event and takes some data as input and produces statistical output. The input data might be anything from a single digit to a table or database.

Here are a number of example of model interfaces:

Single page form
An example might be a simple BMI/BMR (Body Mass Index / Basal metabolic rate) calculator that take in details from a user and offers an result on screen.

Multi page form / wizard
Complex models might take many different parameters as inputs as well as data tables. The resulting model run might not be instant. Our software at Creme Global often fits this description. Inputs are collected in a multistep process.

Single page app
Google Maps is a type of model. You give it input, your destination and, if it works out an ideal route, we can assume that some modelling is going on in the background. Even Google search which uses machine learning is a type of model output.

Excel Macro
Many models sit in Excel. Input might be a new excel table containing a list of values. For example a financial forecast macro is a type of model.

API to API
Imagine financial trading software with a model at its core that communicates with other computers and makes decisions?without human intervention. They say 90% of financial trading is algorithmic these days.

API to Human
You might consider the "people who bought this" recommendation in Amazon a type of automated model output.

The importance of model interfaces to Data Scientists

Data Scientists rarely build input interfaces on top of models. They may have added simple forms to models in data science products like MatLab or MS Access but not often as web applications. This mostly happens through collaboration with a software development team.

Are Data Scientists actively interested in designing interfaces to their models? Here are quotes from my research. I found the exact quotes valuable enough to have printed them in full. Note there is no full consensus.

"I would definitely be interested in adding a user interface to a successful model, as that would ease model usage not only by fellow academics who wouldn't otherwise want to dive into someone's code but also by people who just want to see how the model behaves and what kind of result they can get out of them."

"That situation is very common in academic or scientific environment, in general."

"From my perspective creating a GUI of any type would fall somewhere down the list of what a model developer must do. An interface might be integrated with documentation."

"Simple models/apps could be far easier and documentation might be skipped in favor of the GUI (documentation really is? part of the GUI). Complex team-built models would be a different animal. Many of the complex team products being built today fold documentation into the product, i.e. it is not a separate written document but part of the software itself - code and documentation built at same time. One could include a GUI as part of this process; I know of HTML based systems for the documentation that could be the framework for a GUI."

"I would think post-fitting a simple model might be possible but I wouldn't start to try and retrofit onto a complex model."

"Public funding means sharing, so the we would eventually be interested in publishing and sharing our works so that other people can use it. Of course, there are commercial projects aiming at obtaining patents for the works, but obviously those are not for personal ease of use, either."

Form design for models and data science tool

Good form design is a topic in web usability. When designing input forms for Data Scientists there are extra considerations.

- You will find that complexity of meaning demands you provide good help
- Terminology in forms might can be confusing. For example R terminology has to be learned to be understood.
- Include good validation. For example if a form asks you for a % it should only allow 0-100 range and possibly not decimals.
- Model inputs forms are often slow to complete and might take multiple sessions. Perhaps allow saving. More inputs generally mean a more detailed model.

Reproducibility

In Data Science , Modelling and the sciences in general, reproducibility is a cornerstone topic. As a Designer of data science products you will find that an understanding of reproducibility is very beneficial.

How do you design reproducibility in to data science products? I found a recent paper titled Ten Simple Rules for Reproducible Computational Research a good starting point for a design. The rules:

Rule 1: For Every Result, Keep Track of How It Was Produced
Rule 2: Avoid Manual Data Manipulation Steps
Rule 3: Archive the Exact Versions of All External Programs Used
Rule 4: Version Control All Custom Scripts
Rule 5: Record All Intermediate Results, When Possible in Standardized Formats
Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds
Rule 7: Always Store Raw Data behind Plots
Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
Rule 9: Connect Textual Statements to Underlying Results
Rule 10: Provide Public Access to Scripts, Runs, and Results

The dangers of an oversimplified solution like Excel are further exposed by these rules

Visualisation of Data

How to think about data visualisation

In 2016 I hosted a series of talks on "Design for Data Science" as part of the Predict Conference. One of our speakers Krystian Samp gave an excellent talk titled "Perspectives on design of analytics products" which you can watch online here. He essentially gave 10 thinking tools, or as he calls them, perspectives, for improving an analytics design. Such a design might be a dashboard, report, or any design incorporating analytics

Following are Krystian's 10 perspectives. Krystian makes the point that these are only a selection of perspectives. There are more and you should construct you own also. This is important because companies tend to stick to one perspective. What we are doing here is thinking creatively about design visualisation design for analytics.

1: Base your analytics designs on actual user questions

Review every written or spoken word you have from your customers or relevant voices. For example you might parse online interviews for insight. Once you have built a large map of primary and secondary questions & answers you will have starting point for your analytics design.

2. What behaviour do you want to drive
Certain parts of your data arehigh value. Your analytics design should encourage users towards that data. One design principle you can use to achieve this is hierarchy.

3. Static vs dynamic
If your data does not change it might be interesting to see once, but then it becomes unuseful. Be aware of this when planning your design. Also do you have to do much filtering to make a chart useful? For data that does change often are there meaningful actions you could attach to that data?

4. Facts vs Reasons
In this perspective Krystian makes the point that we tend to list facts in our charts without giving the reasons, and that when we look back at our data the reasons are often in there somewhere. Krystian suggests you "think about what data you have can

explain other data you have". A chart showing a rate of increase might be an explanation but absolute numbers are not.

5. Point of reference
If you are showing values to a user, can you create a point of reference by showing in addition historical data / trends / competitors data, for example. This data is often available to you.

6. Past present & future
Following from our last point, think about our data and time. What do we learn about it over time? Particularly the future is ignored. Can we use predictive technology to forecast what is going to happen next? This often requires a strong learning set of data.

7. What must happen in 5 seconds
In this perspective Krystian is again hinting at hierarchy in visualisation design. If you think about what captures attention over 5 seconds, 60 seconds or even minutes, you have a map of how you might create hierarchy in your design. Does your hierarchy reflect the user's best interests?

8. Aggregate VS what is aggregated
Every visualisation is an aggregation. Choosing what to aggregate is an act of curation. We aggregate because we see a pattern in some data. If there is no pattern, then why aggregate? Also be aware that different data has different characteristics once aggregated.

9. Together or Apart
A table in a database often equates to a visualisation. We don't often cross tables to get to a new insight. We should look for opportunities where this crossing gives exponential returns of insight. It is this crossing of data that often become the famous "exposed by the data" news pieces. For example an increase in GDP might be attributed to manufacturing in one table, but exposed to be caused by property price fluxuations in another. The actual truth here is vitally important.

10. Inform VS Motivate
This point rhymes a little with point 3. If you understand your user and data you will know where you should design to encourage the user to interact.

I hope Krystian's list encourages you to think outside the box when designing analytics interfaces. Your can follow Krystian on twitter here.

Lessons from Edward Tufte

Edward Tufte is a pioneer in the field of data visualization and his books on this such as The Visual Display of Quantitative Information are classics. Here are a few lessons from Tufte to improve your analytics interface designs.

Aim for Graphical Excellence
Tufte defines graphical excellence as "that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space". Notice that tufte differentiates between ideas and data points.

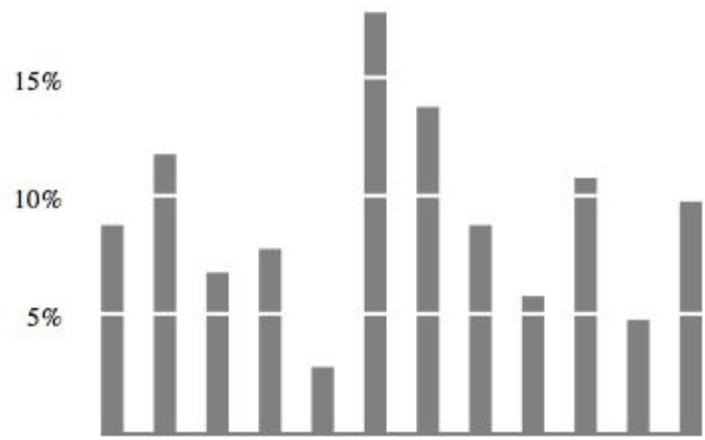
Graphical Integrity
Because Tufte spent time reviewing graphics in newspapers he has strong opinions on misleading graphics. He even devised a formula to describe this called the lie factor:

$$\text{Lie Factor} = \frac{\text{size of effect show in graphic}}{\text{size of effect in data}}$$

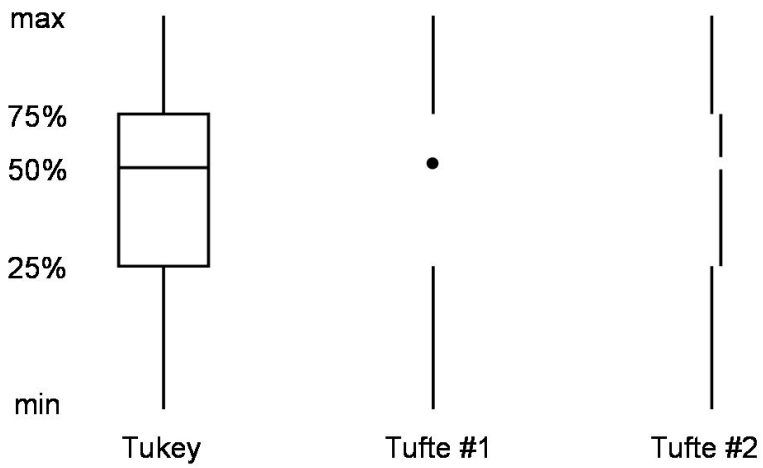
Data Ink Ratio
Tufte is a minimalist. He targets "non data related ink" as a contributor to bad visualisations. Consider what percentage of my ink (pixels) show data?

$$\text{Data-Ink Ratio} = \frac{\text{data-ink}}{\text{total ink used to print the graphic}}$$

His ideas are working well on this bar chart:



Here is Tufte's redesign of the box plot. Perhaps a step too far:



In summary think about removing the stuff that does not change when data changes.

Smallest effective difference
In Tufte's words "Make all visual distinctions as subtle as possible, but still clear and effective". In other words do the least you can do to highlight a difference. Tufte describes this idea as the 'Occam's razor' of information design. For example, do you actually need colour on your bar chart?

A Data Scientist's favourite charts

It is useful for a Designer to know the go-to charts for a Data Scientist. If you are allowing a limited amount of charts in your tool, from my experience I would make at least the following charts available: **Bar, Line, Histogram, Box**.

Understanding the best use and possible configuration of each chart type would be far too expansive a topic for this document. However I can say that personally I have found the documentation of charting libraries very useful in brushing up on theory, and in particular Google Charts.

Choosing a charting library

Be careful when choosing a charting library. Make sure it has all the features you need. It is not ideal for your Dev team to make modifications to a library that soak up time and might become redundant with an update. Don't pick a library just because it is beautiful only to then find out it is missing feature or charts. Don't pick a charting library that is too open and configurable, but not prescriptive enough. In our company we chose Google Charts in the end over HighCharts (beautiful) and D3 (configurable).

In summary, be very involved in your team's choice of a charting library. As a Designer you should be in a position point out cul-de-sacs to the user and business from a bad library choice.

The End