SOME TOOLKIT OPTIONS

This overview is based on awesome GovHack Handbook

Awsome lists of awesomeness

We will list only some tools and libraries in this document, but we're so far from covering the full list of what's out there. So if you're after some tools for a particular programming language, platform, frontend or backend development, and so on check out the list of awesome lists (and try not to be too overwhelmed by awesome projects).

Project management

Honestly, your best project management tool is probably a whiteboard, or paper blutacked to a wall, with different coloured post-it notes. It gives your an immediate, physical, tangible thing to get up and interact with, look at, scribble on, and easily rearrange that no digital system is going to come close to giving you.

But if you really want a digital solution then take a look at Pivotal Tracker, Trello, or Matterhorn.

Code editors

You probably already have a favourite code editor, but if you're looking for inspiration go and check out Atom, Orion, Sublime Text 3, and Brackets.

Data visualisation

Data visualisation encompasses a broad range of fields, techniques, and tools for creating visual representation of data for human consumption. The geographic and tabular data fields have rich toolsets for visualising their particular types of data, so keep on scrolling if you're after some specific tools.

The theory of it all

For advice on the use of colour check out Paul Tol's advice on good colour schemes and the Fink Lab's collection of colour schemes.
The School of Data has a set of data visualization guidelines by Gregor Aisch that are worth a read.
Juice Analytics has good roundup at Data Storytelling: The Ultimate Collection of Resources.

Resources for inspiring

If you're stuck for inspiration check out the Visual Complexity.

Resources for building

If you're not sure exactly what tool you're after and like staring at lists of tools waiting for something to leap out at you then check out this: selection of tools for visualisation

Web visualisation tools

We couldn't mention data vis without giving a nod to D3.js (Data Driven Documents) for creating interactive and amazingly detailed visualisations – find out more about Why D3.js is So Great for Data Visualization. Bewarned though, the learning is quite steep as you're starting out, but the web is full of thousands of D3.js examples that you should have no problems hacking into the shape you want.

Beyond D3.js have a look at Highcharts, Google Charts, jit.

Visualisation as a Service

For some quick out-of-the-box charting plot.lt, Datawrapper, and infogr.am both offer great charting as a service tools for easy prototyping without having to write any code.

Desktop tools

If you're playing with data vis on the desktop you'll find a lot of the tools are commercial in nature, but Tableau is worth a look (as well as the School of Data tutorial Analysing Datasets with Tableau Public).

Android native charting libraries

If you're in need of toolts for building charts and graphs on android have a look at Androidplot, ChartDroid, or achartengine.

Geographic data

Quick and dirty – just show me what the data looks like

The first thing you'll probably want to do when you find data is to actually just quickly view it to see what it looks like, check if the data is what you thought it was, if the geographic distribution is about right, etc.

For really quick and simple viewing you can drop most common sorts of spatial vector data on geojson.io and see a quick representation of it (as well as then exporting it back out to a different format). All of the processing is client-side though, so you might want to avoid giving it a huge or complex dataset. MapStarter is another similar service, though it only allows you to export the data as an image (or a simple web map).
Oh – and did you know that GitHub will render any GeoJSON files that you commit to your repo. Fun!
For any larger or more complex datasets QGIS is a great open source cross-platform tool for viewing any and every type of spatial data.

Converting between data formats

GDAL is a fantastic open source project that has been embedded in a lot of the software in the spatial world. To translate vector data in GDAL reach for the ogr2ogr command (if you're on Windows ogr2gui is available too), for raster (picture) data gdal_translate will convert almost anything to almost anything else.
If command-line tools aren't your thing skip down a bit to the section on QGIS for a cross-platform

Geocoding – turning an address into coordinates

Your geocoding needs will likely fall into one of two categories: Needing to geocode an address provided by the user vs needing to batch geocode a set of addresses in a dataset.

There are some free / open source RESTful APIs for geocoding, which you could happily either wrap a UI around or issue batch requests to yourself. These include the MapQuest Nominatim Search API, the MapBox Geocoding API, and the GeoNames Search API.
If you're after a more set-and-forget geocoding service that will geocode a whole file of addresses with having to fiddle with making your own API calls then take a look at CartoDB's geocoding functionality – and Google Fusion Tables is still kicking around in "experimental" mode (tutorial here).
Lastly, the Python library geopy provides a convenient API wrapper around almost every geocoding service known to humanity.

Analysis

PostGIS is an extension for PostgreSQL providing spatial capabilities for both vector and raster data. In spatial database-land it is unequalled in the sheer range of functions it makes available, their ease of use, and speed (it's written in C).

QGIS – a free and open source cross-platform Geographic Information System with the ability to create, edit, visualise, analyse, and publish spatial information. Thanks to being built on top of GDAL (amongst others) QGIS is capable of reading and writing almost any format of spatial data that you can throw at it – including direct connections to PostGIS databases.

Python

For some general words on working with spatial data in Python check out GIS with Python, Shapely, and Fiona.
For working with vector data check out Shapely (manipulation and querying geometry), Fiona (a Python API into GDAL/OGR), pysal (for spatial analysis).
On the raster side of the equation head straight to Rasterio.
Honourary mentions to pandas (if you need to munge and otherwise play with GeoJSON or CSV files), and cartopy and nodebox-opengl if you need to make pretty pictures or animations.
There's a more complete list of a bunch of other great Python spatial libraries over here that's well worth a read.
We should mention – pretty much anything you can do here you can also achieve with the tools available in a GUI in an application like QGIS.

Java

Java is almost equally as awesome as Python, with a similarly rich ecosystem of libraries and applications (GeoServer the popular spatial data server is primarily Java-based).
For playing with vectors cast your eyes over Spatial4j (general purpose geospatial data library), JTS (Java Topology Suite) (do things with geometry!), or Apache SIS
For everything and anything check out GeoTools – the Swiss Army Knife of spatial in Java-land for reading/querying/analysing/rendering vector and raster spatial data.

R (Arrr!)

As a primer you should check out Starting Analysis and Visualisation of Spatial Data with R.

Surprise! There's actually a great StackExchange question on this very topic. In addition to the resources listed therein, James Chesire has a great (and quite accessible) write-up on his blog at R Spatial Tips. Robin Edwards also has some great words and examples about 3D Mapping in R.
And there's also spatstat if you want to delve down into spatial statistics and analysis.

.NET

You'll find pretty reasonable support for spatial data in .NET-land with the likes of:

Geo – a powerful little .NET 4.0+ library for querying and manipulating vector data.
NetTopologySuite – a port of the aforementioned popular JTS (Java Topology Suite) library for querying and analysing vector data.
SharpMap – a geo app framework for vector and raster data that includes its own rendering engine.
MapWindow – an all in one desktop GIS tool + an ActiveX control for mapping + a C# library for handling vector data.

Ruby

Your options are not quite as rich here – but have a look at geokit, georuby, and RGeo.
Daniel Azuma's series of blog posts on doing geospatial in Ruby is going to be worth your time.

A few other tools

In recent times a few really handy and modern little web tools have popped up for doing simple and/or common tasks with spatial data.

geojson.io for quickly and easily creating, viewing, and sharing vector data as GeoJSON (and other common formats).
Ogre as a web client to the ogr2ogr utility in GDAL. Easily convert between vector formats!
GIS Convert for easily converting between spatial and spatial-like formats.
GeoGig if you want to apply the principles of Git to spatial data.
epsg.io if you've found some data but it's not in a standard projection (e.g. latitude and longitude, web mercator) then find the "EPSG" code and stick it in here to find out more about it.
GitSpatial if you just want to wrap a spatial API around your GitHub-hosted GeoJSON data.
TopoJSON an extension for GeoJSON that encodes topology tl;dr it'll make your GeoJSON up to 80% smaller.
And an honourary mention to Shape2Earth for allowing the easy creation of maps for Google Earth.

Storyboarding and making videos

For screen recording check out software like ActivePresenter that will allow you to record demos of your action.
For mixing clips together the YouTube Video Editor is very handy; though VLC or LWKS may also be useful.
Videos can be uploaded to YouTube, Vimeo, etc