Index

A

Access, command line, 230–232
  append, files, 231
  availability of, 231
  chain together commands, 231
  count, number of words and lines in file, 231
  download web pages, 231
  edit large text files, 231
  examine, beginning or end of file, 231
  find patterns, 231
  find, duplicate rows in file, 231
  join, files horizontally, 231
  list files and directories, 231
  sort data files, 231
Analytics code. *See* Analytics coding
Analytics coding, 29, 78, 126, 137
  break up data flows into data steps, 85–86
  clean data minimum of locations in data flow, 93–94
  cleaning a data field, keep original raw field, 94–95
  clearly label running order of code files, 83–84
  consistent cleaning, 126
  create unique identifier for records, 98–99
  data provenance, 91
  don't jump in and out, code file, 86–87
  drop datasets, start of code execution, 84–85
  examples, 76–78, 91
    casting, 91
    cleaning, 91
    CSV data file, 76
    De-duplicating, 91
    deriving, 91
    enriching, 91
    filtering, 91
    human-keyed inputs, 78
    joining, 91
    multiple code files, 78
    multiple environments, 78
    multiple outputs and output formats, 78
  favor code, not embedded in proprietary file formats, 81–82
  filter data with flags, not deletions, 95
  Guerrilla analytics workflow, 76, 77, 91, 92
  identify fields with metadata, 96–97
  log code execution, 87–89
  pitfalls and risks, 78–79, 92
    difficulty in testing, 92
    discarded records, 92
    failure, distinguish data from original data, 92
    loss of data record provenance, 92
    lost data provenance, 78
    overwrites, 92
    reproducibility, impaired, 78
    time wasted in review and development, 78
  produce clearly identifiable data outputs, 80
  rename data fields with field mapping, 99–100
  repeated cleaning, 126
    dataset of country names, 126
  repeated joining, 126
  use one code file per data output, 79–80, 81
  write code that runs from start to finish, 81
Analytics environment, 61
Analytics pre-sales and proof of concept, 13
Analytics projects challenges, Guerrilla analytics, 18–22
  data generated by people, 20
  external data, 20
  foreign data environment, 21
  reskilling, team quickly, 22
  testing and reviewing analytics, 21
  tracking multiple data inputs, 19
    front-end reports, 19
    relational database, 19
    spreadsheets, 19
    unstructured data sources, 19
    web-scraped data, 19
  tracking multiple data work products, 20
  upskilling, team quickly, 22
  version control of analytics, 20
  versioning multiple data inputs, 19

Analytics risks
  changing of raw data, 27
  conflicting changes, datasets, 26
  failure, communicate updates to team knowledge, 28
  fragmented code, cannot be executed, 29
  impact of failure to address, 31
    analytics project, 31
    effect on team efficiency, 31
    environment inefficiency, 31
    team analytics outputs, 31
  inability
    to easily navigate, analytics environment, 25, 26
    identify source of dataset, 30
    reproduce work products, 25
  lack of clarity around derivation of analysis, 30
  losing link
    data received and storage location, 23, 24
    raw data and derived data, 25
  multiple copies of files and work products, 28
  multiple versions of tools and libraries, 30
  out of date documentation misleads, team, 27
Analytics team outputs and efficiency
  pitfalls and risks, 127–128
    copy and paste code, 127
    data populations, 128
    inconsistent work products, 127
    reduced efficiency, building data from scratch, 127
    reduced efficiency, lack of abstraction, 127
Analytics testing, five Cs, 242
  aCcountability, 242
  coherence, 242
  completeness, 242
  consistency, 242
  correctness, 242
Audit time-stamped record of code, 89
Automated code documentation tools, 229
Automation, code file execution, 84

B

Back-end data extractions, 43
Better analytics testing, five Cs, 242
Bug tracking, 221
Build tests, 156, 178–182
  categories of, 192
  organizing code, 185–187
    make test code easily identifiable, 186–187
    make test code part of build code base, 187
    use one build test code file per test, 185
  types of, 178
    medium tests, 178
    small tests, 178
Build tool, 228, 233
Builds, consolidating knowledge in
  centralized and version-controlled program code, 130

C

Casting, dataset, 91
Centralized and controlled file system, 236
Changing business rules
  analytics projects, challenges, 20
Checksum calculation, record of data field, 160
Code documentation, 237–238
  automatic code documentation tools
    Doxygen excel, 238
    Javadoc, 238
Code files
  muddled and clear running order, 84
Code libraries
  for data wrangling, 228
  for machine learning and statistics, 228
Code that manipulates data
  version control changes, 36
    addition of cleaning rules, 36
    change in business logic, 36
Comma separated value (CSV) file, 47, 76, 211
Command line, 227
  data profiling, 227
  tools, 71
Common routines applied to data
  version control changes, 36
Communication, Guerrilla analyst, 212–213
Complex folder structures, 56
Consistency tests, 174
Consolidate team knowledge in version-controlled builds, 37
Convenience datasets, 130
Core capabilities, 238
  access, command line, 238
  common code libraries, 239
  DME, 238
  encryption, 239
  file system, 239

high-level scripting language, 238
revision control, 238
visualization, 239
Creating analytics, reproducible
analytics projects, challenges, 21
CRISP-DM reference model, 16, 17
CSV file. *See* Comma separated value (CSV) file

D

Dashboard tools, 211
Data
analysis and visualization, 102
cleaning layer, 141
coherence, 168–169
corruption, 68
dashboards, 211
description, 221
dynamic workflow. *See* Dynamic data
factory, 7
field mapping, 99
folder, 63
generated by people, 20
gymnastics, 210
integration, 102
journalism, 12. *See also* Data journalism
lab, 7
loads. *See* Data load
log, 63
manipulation languages SQL, 71
manipulation, environment, 82, 241. *See also* Data manipulation environment (DME)
mapping structure, 99, 131
model, efficiency of, 184
preparation, 68
profiling, 102
team, work products cannot identified, 120
testing, 102, 156. *See also* Testing data
UID, 63
validation, 50
Data analytics
definition, 4, 5
environment, 14
examples, activities, 5–6
for research
projects example, Guerrilla analytics, 12
projects types, 6–7
Data builds, 132–141, 175
data flows out of raw, 133
illustrative example, 175–176, 178
practice tips
automate execution of build code, 137
build version in the build dataset names, 135
builds do not produce work products, 139
create documentation datasets, 139–140
decouple raw data from build data, 134
develop layered build code, 140–141
embed reporting in build, 138–139
generate data builds, version-controlled code, 134–135
tear down and rebuild, 135–136
testing in build data flow, 137–138
structure of, 133, 175, 176
ad-hoc data sources, 175
intermediate and interface data location, 133
raw data location, 133
temporary dataset location, 133
version-controlled build code, 133
version-controlled code files, 175
Data extraction, 43
front-end data extractions, 43
optical character recognition (OCR), 44
pitfalls and risks, 44–45
extracting from a moving target, 44
front-end reports cannot be reproduced, 45
inability to assess completeness, 45
no checks of OCR data, 45
no record of web pages, 45
no reproducibility of extraction process, 45
nothing to compare to, 45
practice tips
calculate checksums before data extraction, 48–49
capture front-end reports, 49–50
consistency check OCR data, 52
extract data, agreed file format, 46–47
freeze source system during data extraction, 45–46
save raw copies of scraped web pages, 51
Data factory, 7, 13
Data lab, 7, 8
Data load, 54, 66, 67, 70, 71–72, 74
challenges, 74
data folder, 66
Guerrilla analytics environment, 67
preparatory work, 67
validating data load, 67

Data load *(cont.)*
  Guerrilla analytics workflow, 16–17, 18, 43, 44, 54, 66, 67, 76, 77, 91, 92, 102, 103, 116, 117, 131, 149, 150, 158, 159, 218, 241
  pitfalls and risks, 68
    data corruption, 68
    data preparation, 68
    link, data on file system and loaded data, 68
  practice tips
    data load preparations on copy of raw data files, 70
    load data as plain text, 73–74
    prefer one-to-one data loads, 71–72
    preserve the raw file name and data UID, 72–73
  raw data, 66
  relational system extract, 66
  semistructured data, 66
  validating, 67
Data manipulation environment (DME), 13, 43, 66, 195, 227, 229
  domain-specific tools, 229
  mapping, received data, 23
  NoSQL databases, 13, 229
  programming language, 229
  statistical environments, 13
  structure, 105
Data on web pages, 228
Data provenance, 12, 32–33, 36, 40, 69, 91, 101, 113
  changing data, affects, 94
  data modified, 120
  difficulty in testing, 101
  discarded records, 101
  overwrites of data, 101
  record provenance, 101
  work products cannot identified, 120
Data receipt
  folder structure, 64
  guerrilla analytics workflow, 54, 55
  pitfalls and risks, 54–56
    data lost on file system, 54
    local copies of data, 54
    multiple copies of data exist, 54
    original data, renamed, 55
    supporting information, not maintained, 54
  practice tips
    create unique identifiers, 59
    keep supporting material near data, 61–62
    never modify raw data files, 60–61
    single location, 57–59
    store data tracking information in data log, 60
    version-control, 62–63
  receiving data, 54
Data security, 228
Data testing, 158
Data wrangling, 207
Data/people feedback loops, 104
Databases, 13, 20, 48, 160, 209, 233
  variety of, 43
Datasets, 14
  clean and raw data fields, 94
  general data structure, 14
  naming conventions, 97
  test input, 187
  test register, 187
  test result, 188
  without naming conventions, 96
Date field, 123
Date-cleaning code, 142
Debugging intermediate datasets, 86
Decouple data, 59
DME. *See* Data manipulation environment (DME)
Dynamic data
  workflow, Guerrilla analytics, 18

E

Encryption, 228, 235
  encrypting media, 235
  moving data portable media, 235
End-to-end code, 87
Environment
  Guerrilla analytics, 90
External data
  analytics projects, challenges, 20
External software and libraries
  version control changes, 36
Extracted data, journey of, 48
Extracted file format, 46

F

Field mapping approach, 100
Front-end data extractions, 43
Front-end report, 19, 45, 49–50, 51

G
Generators, program code files, 104
Guerrilla analytics, 33
  analytics work, 35
  applying, 38–40
  data flowing, 36
  definition, 9–11
  link, data on file system and data in analytics environment, work products, 36
  manipulate data, 35
  prefer automation, program code, 35
  project-specific rules, 34
  space, cheap/confusion expensive, 33
  structures over heavily documented, 34
  visual project structures, 34, 35

H
Hash function, 98, 162, 163
High-level scripting language, Guerrilla analytics, 232

I
In-line documentation approach, 124
Invalid values for correctness tests
  allowed values, 165
  invalid ranges, 166

K
K-fold cross validation, 200

L
Large tests for build tests, 181–182
Loading data, 74

M
Machine learning and statistics libraries, 3, 235
  data science, 236
  general programming language, 236
  statistical modeling, 235
Management approach, applying principles
  controlling complexity with project folders, 38
  controlling delivery with process, 39
  pushing administration down to analysts, 39
Management information (MI), 77, 114
Marked-up data, 102
Maths and stats
  choice of variables, 213
  dependent variables, 213
  independent variables, 213
  performance, 213
  testing, 213
  transformations, 213
Medium tests for build tests, 180–181
Metadata, 59, 75, 96, 155, 164, 164–165
MI. *See* Management information
Moving data sources
  analytics projects, challenges, 20

N
NoSQL, 4, 14, 66, 229

O
OCR. *See* Optical character recognition
Optical character recognition (OCR), 44, 52
Ordinary work products, 194–198
  5C tests, 195
    aCcountability, 195
    coherence, 195
    completeness, 195
    consistency, 195
    correctness, 195
  testing accountability, 198
  testing coherence, 197–198
    work product coherence test, 197
  testing completeness, 195–196
    checksums, 195
    comparing population in dataset, 195
    correct population of data, 195
  testing consistency, 196
  testing correctness, 196
    distribution of data, 196
Orphaned archives, 56
Overall documentation, for code documentation, 237

P
Parse file formats, 232
Performing data cleaning, 94
Personal folders, 56
Profiling for correctness for tests, 166
  blank fields, 166
  fields with only one value, 166
  fields with unexpected number, 166
  high-frequency values and low-frequency values, 166

Programming, 210–211
    append files, 211
    character encoding, 211
    compression, 211
    create samples, 211
    data manipulation, 210
        data gymnastics, 210
        design short data flows, 210
        domain-specific language, 210
        reshape data, 210
    data visualization language, 211
    find and replace, 211
    find patterns, 211
    Iterate through files, 211
    sort contents, 211
    split files, 211
Programming languages, 87, 103, 241
Projects example, Guerrilla analytics, 11
Proprietary graphical inputs, 82
Provenance
    analytics code, preserves data, 91
    analytics projects, lack of data, 240
    analytics work, 36
    break up data flows, 85
    breaking data, 48
    challenges of data, 224
    clearly label, running order of code files, 83–84
    creating, unique identifier (UID), 98
    data
        modified, after leaving team, 120
        teams, work products cannot identified, 120
    data preparation, 68
    data receipt and load, 218
    Dave's bar of data, 223
    discipline, maintaining data, 217
    Guerrilla analytics environment, 48
    Guerrilla analytics workflow, 66
    impact on loss of data, 34
    impact, data manipulations, 86
    implications for data, 35
    labeled versions for data, 228
    less traceable data, 69
    log all work products, 114
    lost data, 78, 92
    maintain data, 32–33
    pitfalls and risks, analytics code, 92
    prefer one-to-one data loads, 71
    source code control, maintenance of data, 215
    testing data, 198
    visualization, 232
    work product, 104

R

Raw data
    column, 96
    layer, 141
    version control changes, 36
Real-world data, 163, 240, 242
Referential integrity tests, 174
Relational system extract, 66
Reporting
    analytics work products, 125
    challenges for traceability, 119
    collaboration with business user, 116
    components, 119, 120
        analytical components, 119
        figures, 125
        tables, 125
        text, 125
        written components, 119
    examples, 118
    in-line documentation, 124
    pitfalls and risks, 120–121
        consistency, 121
        data provenance, 120
            data modified after leaving team, 120
            data team, cannot be identified, 120
        implications, Guerrilla analytics team, 121
    practice tips
        create one work product per report component, 122–123
        liaise with report writers, 122
        make presentation quality work products, 123–124
    risks, 125
        consistency of report components, 125
        data modified, after leaving, 125
        presentation quality work products, 125
        work products not identified, 125
    workflow, 116, 117
Risks, 22–30
    changing of raw data, 27
        effect, 27
    conflicting changes to datasets, 26
    failure, communicate updates team knowledge, 28
    fragmented code, cannot executed without author's input, 29

inability, identify source of dataset, 30
lack of clarity around derivation analysis, 30
use of multiple tools, 30
link, data received and storage location, 23–24
link, raw data and derived data, 25
code files, 25
derived data for work product, 25
multiple copies of files and work products, 28
multiple versions of tools and libraries, 30
navigate, analytics environment, 25–26
data cleaning, 26
end-to-end data flow, 26
out of date documentation misleads team, 27–28
reproduce work products, disappeared source datasets, 25

S
Semistructured data, 66
Service builds, 142–145
practice tips
automate service deployment, 145
embed service build version in service names, 144
generate service builds with version-controlled code, 143–144
locate and deploy services, 144–145
tear down and rebuild services, 145
program code, 142
Skills, 208–210
communication, 209
data environment, 209
hacking or programming, 208
math and statistics knowledge, 209
mindset, 209
software engineering, 209
substantive expertise, 208
visualization, 209
Small tests, for build tests, 178–179
Social network analysis, 4
Software engineering, 6, 140, 151, 210, 215–216, 219, 222, 232, 242
problems, workflow management, 215
source code control, 215–216
Source code control, 229
availability of revision control, 229
experimental copy of code, 229
level of training and expertise, 230
subfield of software engineering, 230
technology, 229
tracks, program code, 229
Sources of variation, for data understanding, 129
changes in raw data, 129
changes to code, 130
changes to common libraries, languages, and services, 130
changes to input parameters, 130
nondeterministic code, 130
Statistical models, 86, 157, 194
Subfolder structure, 66
Substantive expertise, 212
System data extractions, 43

T
Test code development, 182–184, 192
practice tips
create common test results dataset structure, 183, 183–184
develop modular build code help testing, 184
use common layout in test, 182–183
Test data, organizing
input datasets, 187
intermediate datasets, 188
practice tips
automate build test execution and reporting, 191
establish utility data structures in setup, 189–191
retain intermediate test datasets, 188
store test results and test failure details, 189
result datasets, 188
Test execution, 171, 183, 191
Testing, 18
builds. *See* Build tests
challenge of, 152–153
completeness, 153
scale, 152
unknown data, 152
code, 151
comparing expected and actual, 151–152
correctness of work products, 149
data. *See* Testing data
detect defects, 149
models. *See* Testing models, general tips
practice tips
establish testing culture, 153–154
give each test unique identifier, 155
organize test data, test UID, 156

Testing *(cont.)*
test early, 154
test often, 154–155
work products. *See* Testing work products
workflow, 149, 150
Testing and reviewing analytics, 21
Testing data, 150
accountability, 170
coherence, 168–169
data relationships, 168
relationships different from expectations, 169
table relationships, 169
testing incomplete relationships, 169
completeness, 160–163
approach to unstructured data, 162
approach, structured data, 160–161
correct ranges of data, 160
extracting data, 160
practice tips
agree and test the hash function, 162
agree approach blank data, 161–162
allow a threshold difference for checksums, 162
capture checksums before data extraction, 161
consistency, 167–168
correctness, 163–164
five C'S of
accountability, 159
coherence, 159
completeness, 159
consistency, 159
correctness, 159
workflow, 158, 159
analytics team, key stages, 158
Testing models, general tips, 200–201
Testing statistical models, 199–200
Testing work products, 156

U
UID. *See* Unique identifier
Unique identifiers (UIDs), 59, 62, 72–73, 98, 105–107
Unstructured data extract, 68

V
Version control, 242
Version control of analytics
analytics projects, challenges, 20–21
Version-controlled build code, 133
Version-controlled datasets, 100, 142
Versioning multiple data inputs, 19
Visualization, 214, 232–233

W
Work products
classifying, 224
data provenance, 224
internal or external delivery, 224
project work stream, 224
creating and tracking in Guerrilla analytics environment, 114
essence of, 104
data sources, 104
generators, 104
outputs, 104
folder structure, 109
generation code, 108
linking, 223
pitfalls and risks, 104–105
practice tips
avoid clutter in DME, 110–111
avoid clutter in file system, 108–110
log all work products, 114
track work products, unique identifier (UID), 105–107
use, convention to name complex outputs, 113
version control work products, 111–112
work product generators and outputs, 107–108
UID in file system and DME, 106
workflow, 102, 103
Workflow management, 217, 218, 236
common information, 220–221
completing, work product, 220
creating, work product, 220
review, work product, 220
tracking and understanding, work product, 221
common states and transitions, 219–220
granularity, 224
in analytics, 217
levels of review, 222
software engineering tools, 217, 236
specific information for builds, 221–222
bug tracking, 221
change control, 222
related development, 222
version control, 222
specific information, data receipt and load, 221

data description, 221
data storage and load, 221
delivery details, 221
related work, 221
types of, 218–218, 219
usage, 224–225
geographically dispersed teams, 225
importance of traceability, 225
life of project, 225
project dynamism, 225
team experience, 225
team size, 224