

Python Data Cleaning PDF Page

Python Data Cleaning PDF Page: Content + <head> (Uncodemy)

FINAL URL: <https://uncodemy.com/python-data-cleaning-pdf>

Publish this page at exactly this URL (non-www, no trailing slash). Canonical, og:url and sitemap entry must match it.

Prepared: 26 October 2026 · **Primary keyword:** python data cleaning pdf · **Search volume:** not checked (keyword tool out of credits); verify in Search Console.

Read first (5 flags):

1. **Your course PDF title, page count and size were not shared.** Download cards are fill-in slots ([CONFIRM]). Card 1 is your course PDF; cards 2 and 3 only if you have more files.

2. **One unique file per button.** The same course PDF repeated on many pages is fine as a resource, but each page's text must stay unique and useful without the file.

3. **Free or gated?** Not stated. The page says "Download", not "Free". If a form is used, add a consent checkbox and Privacy Policy link (DPPD Act 2023).

4. **No made-up stats or testimonials;** no placement or salary claims without a named, dated source.

5. **Author and reviewer are placeholders;** use real named trainers and one real "Last reviewed" date.

PART 1: Page plan

Item	Decision
Final URL	https://uncodemy.com/python-data-cleaning-pdf
Intent (anti-cannibalisation)	CLEANING WITH PYTHON AND PANDAS. SQL cleaning and the tool-neutral process are on separate pages.
Why not thin	Explainer-style sections, FAQs and cross-links make the page useful even without a download
Schema	WebSite, Organization, Breadcrumb, CollectionPage, FAQPage (only FAQs visible on the page). PDF ItemList provided commented, to enable after titles are final
Share image	https://uncodemy.com/image/logo-og.jpg for now; custom 1200x630 card recommended

PART 2: Page content (copy-paste ready)

SEO Title (35 chars):

Python Data Cleaning PDF | Uncodemy

Meta Description (133 chars):

Download the Python data cleaning guide PDF from Uncodemy: missing values, duplicates, types, text and outliers with pandas examples.

H1: Python Data Cleaning PDF Downloads **Breadcrumb:** Home > Python Data Cleaning PDF **Final URL:** <https://uncodemy.com/python-data-cleaning-pdf>

Byline: Prepared by the Uncodemy trainer team [CONFIRM: author name and role] · Reviewed by [CONFIRM: reviewer name and role] · Last reviewed: [DATE]

Quick Answer

This page offers Uncodemy's Python data cleaning guide as a PDF download. It shows how to detect and fix missing values, duplicates, wrong types, messy dates, inconsistent text and outliers using pandas, and how to wrap the steps into a function with checks so that the cleaning is repeatable. The examples build a small messy table in code.

Download Python Data Cleaning PDF

[CONFIRM: free or gated, and whether details are needed before download]

PDF 1 (main): [CONFIRM TITLE: your Data Analytics course PDF]
[CONFIRM: one line on what it covers and who it is for] · [N] pages · [X] MB · PDF · Updated [DATE]
[Download PDF] → <https://uncodemy.com/pdf/python-data-cleaning.pdf> *(your course PDF, uploaded under this file name)*

PDF 2 (optional): [CONFIRM TITLE]
Only if you have a second PDF that genuinely fits this topic. **[Download PDF]** → <https://uncodemy.com/pdf/python-data-cleaning-2.pdf>

PDF 3 (optional): [CONFIRM TITLE]

Same rule. **[Download PDF]** → <https://uncodemy.com/pdf/python-data-cleaning-3.pdf>

Each button must open its own file. Do not point several buttons at one file. Delete unused optional cards.

Cleaning as code

When data cleaning is written as code rather than done by hand, three good things happen. The steps are documented, because the code is the record of what changed. They are repeatable, because the next file can be cleaned by running the same script. And they are reviewable, because a colleague can read the rules and challenge them. The price is a little more discipline: you must decide each rule explicitly instead of fixing things as you notice them. The example below uses a small messy customers table so that you can run every step and see the effect.

A messy example table

The table has duplicates, inconsistent text, missing values and dates in text form.

```
import pandas as pd
import numpy as np

raw = pd.DataFrame({
    "customer_id": [1, 2, 2, 3, 4, 5],
    "name":        ["Asha ", "ravi", "ravi", "MEERA", None, "Kabir"],
    "city":        ["delhi", "Noida ", "Noida ", "NEW DELHI", "Pune", None],
    "signup_date": ["2026-01-05", "05/02/2026", "05/02/2026", "2026-03-10", "bad date", "2026-04-01"],
    "spend":        ["500", "1,200", "1,200", None, "300", "-50"],
})
print(raw)
print(raw.isna().sum())
print(raw.duplicated(subset=["customer_id"]).sum(), "duplicate ids")
```

Text and duplicates

Standardise text, map known variants to one label and remove duplicates deliberately, keeping the first row for each id.

```
df = raw.copy()

df["name"] = df["name"].str.strip().str.title()
df["city"] = (df["city"].str.strip().str.title()
              .replace({"New Delhi": "Delhi"}))

df = df.drop_duplicates(subset=["customer_id"], keep="first")
print(df)
```

Types, dates and numbers

Convert numbers stored as text, parse dates and flag impossible values rather than silently deleting them. `errors=coerce` turns failures into missing values, which you can then count.

```
df["spend"] = pd.to_numeric(df["spend"].str.replace(",", "", regex=False), errors="coerce")

df["signup_date"] = pd.to_datetime(df["signup_date"], errors="coerce", dayfirst=False, format="mixed")

df["spend_invalid"] = df["spend"] < 0
df.loc[df["spend_invalid"], "spend"] = np.nan

print(df.dtypes)
print("unparsed dates:", df["signup_date"].isna().sum())
```

Missing values: decide, then document

After conversion you will have missing values from the original blanks and from values you could not convert. Choose a rule for each column. For a name, a label such as Unknown keeps the row. For spend, the median is a common neutral fill for skewed values, but if the amount drives a calculation you may prefer to leave it missing and report it. Whatever you do, create a flag column or keep a count, so downstream users know which values were filled. A cleaning step that cannot be seen is a source of mistrust later.

Filling and flagging

Fill labels, flag the filled spend values and report how many were affected.

```
df["name"] = df["name"].fillna("Unknown")
df["city"] = df["city"].fillna("Unknown")

df["spend_was_missing"] = df["spend"].isna()
df["spend"] = df["spend"].fillna(df["spend"].median())

print(df[["customer_id", "name", "city", "spend", "spend_was_missing"]])
print("filled:", int(df["spend_was_missing"].sum()))
```

A reusable cleaning function with checks

Wrap the rules in a function and add assertions, so a bad input fails loudly instead of producing wrong numbers.

```
def clean_customers(raw):
    df = raw.copy()
    df["name"] = df["name"].str.strip().str.title().fillna("Unknown")
    df["city"] = (df["city"].str.strip().str.title()
                  .replace({"New Delhi": "Delhi"}).fillna("Unknown"))
    df = df.drop_duplicates(subset=["customer_id"], keep="first")
    df["spend"] = pd.to_numeric(df["spend"].str.replace(",", "", regex=False), errors="coerce")
    df.loc[df["spend"] < 0, "spend"] = np.nan
    df["spend"] = df["spend"].fillna(df["spend"].median())
```

```

assert df["customer_id"].is_unique, "ids must be unique"
assert df["spend"].ge(0).all(), "negative spend remains"
return df

clean = clean_customers(raw)
print(len(raw), "->", len(clean), "rows")

```

Outliers: what to do

Situation	Suggested action
Looks like a typing error (extra zero)	Correct from the source or set to missing, with a note
Genuine rare event (a very large order)	Keep; report medians or analyse separately
Impossible value (negative age)	Mark invalid and investigate
Unknown cause	Keep, flag and ask the data owner

Good cleaning habits

- Never overwrite the raw file; read it and write a new cleaned file.
- Print counts before and after each step.
- Keep rules in one function or script, in version control.
- Add tests for the rules that matter, such as unique ids.
- Write a short log of what changed and why.

Common mistakes

- **Dropping rows with any missing value** and losing most of the data.
- **Filling with zero** when zero means something different from unknown.
- **Parsing dates without checking the day-month order**, which swaps values silently.
- **Chained assignment** that changes a copy instead of the original.
- **No validation** after cleaning.

Testing your cleaning code

Cleaning rules deserve tests just like any other code. Create a tiny table that contains the exact problems you expect, such as a duplicate id, a negative amount and a date in the wrong format, run your function on it and assert that the output has the properties you want. When a new problem appears in real data, add it to the test table first and then fix the code. Over time this builds a safety net, so changes to one rule do not silently break another and so a colleague can trust the function without reading every line.

Checks to run after every cleaning job

Check	Code idea
Row count explained	<code>print(len(raw), len(clean))</code>
Keys unique	<code>assert clean['customer_id'].is_unique</code>
No impossible values	<code>assert (clean['spend'] >= 0).all()</code>
Remaining missing values	<code>print(clean.isna().sum())</code>
Types as expected	<code>print(clean.dtypes)</code>

Related Uncodemy resources

- [Data Analytics Data Cleaning Guide PDF](#)
- [SQL Data Cleaning Guide PDF](#)
- [Python Pandas for Data Analysis PDF](#)
- [Data Analytics EDA Guide PDF](#)

Learn with a trainer [use only the line that is true]

- **Data Analytics course:** <https://uncodemy.com/course/data-analytics-training-course-in-noida>. URL verified on the live site.

(One soft call to action: course link or free demo. No pop-up.)

About these PDFs

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to [\[CONFIRM: feedback_email\]](#) and the trainer team will review it.

Frequently Asked Questions

How do I clean data in Python?

Use pandas to standardise text, remove duplicates, convert types and dates, handle missing values and outliers with documented rules, and validate the result with checks.

How do I handle missing values in pandas?

Detect them with `isna`, then drop, fill with a justified value or flag them, depending on the column's meaning. Record what you did.

How do I convert text dates in pandas?

Use `pd.to_datetime`, with errors set to coerce to expose unparseable values, and check the day-month order carefully.

How do I make cleaning repeatable?

Put the rules in a function or script, add assertions for key checks and keep it under version control.

Should I delete outliers?

Not automatically. Investigate whether they are errors or real events and treat them accordingly.

Are these PDFs free to download?

[CONFIRM: state clearly whether the PDFs are free, or whether a short form is needed first.]

The last FAQ contains [CONFIRM], so it is NOT in the FAQ schema. Fill it and keep it on the page, or delete it.

PART 3: `<head>` section (copy-paste ready)

Final URL: <https://uncodemy.com/python-data-cleaning-pdf>

```
<!-- =====
UNCODEMY <head> | FINAL URL: https://uncodemy.com/python-data-cleaning-pdf
Prepared 26 October 2026. Paste into the page <head>; keep your site-wide favicon, fonts, analytics.
BEFORE GO-LIVE: (1) set dateModified in the JSON-LD to the real publish date;
(2) replace og:image with a 1200x630 share card when available;
(3) once PDF titles are final, uncomment the ItemList block at the bottom.
===== -->

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Python Data Cleaning PDF | Uncodemy</title>
<meta name="description" content="Download the Python data cleaning guide PDF from Uncodemy: missing values, duplicates, types, text and outliers with pandas
examples.">
<link rel="canonical" href="https://uncodemy.com/python-data-cleaning-pdf">
<meta name="robots" content="index, follow, max-snippet:-1, max-image-preview:large, max-video-preview:-1">
<meta name="author" content="Uncodemy">
<meta name="publisher" content="Uncodemy Edutech Pvt. Ltd.">
<meta name="language" content="en-IN">
<link rel="alternate" hreflang="en-in" href="https://uncodemy.com/python-data-cleaning-pdf">
<link rel="alternate" hreflang="x-default" href="https://uncodemy.com/python-data-cleaning-pdf">

<!-- Open Graph -->
<meta property="og:type" content="website">
<meta property="og:site_name" content="Uncodemy">
<meta property="og:locale" content="en-IN">
<meta property="og:url" content="https://uncodemy.com/python-data-cleaning-pdf">
<meta property="og:title" content="Python Data Cleaning PDF | Uncodemy">
<meta property="og:description" content="Download the Python data cleaning guide PDF from Uncodemy: missing values, duplicates, types, text and outliers with
pandas examples.">
<meta property="og:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta property="og:image:alt" content="Uncodemy Python Data Cleaning PDF study material">

<!-- X (Twitter) -->
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:site" content="@uncodemy">
<meta name="twitter:title" content="Python Data Cleaning PDF | Uncodemy">
<meta name="twitter:description" content="Download the Python data cleaning guide PDF from Uncodemy: missing values, duplicates, types, text and outliers with
pandas examples.">
<meta name="twitter:image" content="https://uncodemy.com/image/logo-og.jpg">
<meta name="twitter:image:alt" content="Uncodemy Python Data Cleaning PDF study material">

<!-- Structured data: one @graph (WebSite, Organization, Breadcrumb, CollectionPage, FAQPage) -->
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@graph": [
    {
      "@type": "WebSite",
      "@id": "https://uncodemy.com/#website",
      "url": "https://uncodemy.com/",
      "name": "Uncodemy",
      "publisher": {
        "@id": "https://uncodemy.com/#organization"
      },
      "inLanguage": "en-IN"
    },
    {
      "@type": "EducationalOrganization",
      "@id": "https://uncodemy.com/#organization",
      "name": "Uncodemy",
      "legalName": "Uncodemy Edutech Pvt. Ltd.",
      "url": "https://uncodemy.com/",
      "logo": {
        "@type": "ImageObject",
        "url": "https://uncodemy.com/uc-layout/img/Logo.webp"
      },
      "email": "info@uncodemy.com",
      "identifier": {
        "@type": "PropertyValue",
        "name": "CIN",
        "value": "U85499UP2024PTC195902"
      },
      "contactPoint": [
        {
          "@type": "ContactPoint",
          "telephone": "+91-9818366550",
          "contactType": "admissions",
          "areaServed": "IN",
          "availableLanguage": [
            "en",
            "hi"
          ]
        }
      ],
      "sameAs": [
        "https://x.com/uncodemy",
        "https://www.facebook.com/uncodemyofficial/",

```

```

    "https://www.linkedin.com/company/uncodemy/",
    "https://www.instagram.com/uncodemyofficial/"
  ],
  {
    "@type": "BreadcrumbList",
    "@id": "https://uncodemy.com/python-data-cleaning-pdf#breadcrumb",
    "itemListElement": [
      {
        "@type": "ListItem",
        "position": 1,
        "name": "Home",
        "item": "https://uncodemy.com/"
      },
      {
        "@type": "ListItem",
        "position": 2,
        "name": "Python Data Cleaning PDF",
        "item": "https://uncodemy.com/python-data-cleaning-pdf"
      }
    ]
  },
  {
    "@type": "CollectionPage",
    "@id": "https://uncodemy.com/python-data-cleaning-pdf#webpage",
    "url": "https://uncodemy.com/python-data-cleaning-pdf",
    "name": "Python Data Cleaning PDF Downloads",
    "description": "Download the Python data cleaning guide PDF from Uncodemy: missing values, duplicates, types, text and outliers with pandas examples.",
    "inLanguage": "en-IN",
    "isPartOf": {
      "@id": "https://uncodemy.com/#website"
    },
    "publisher": {
      "@id": "https://uncodemy.com/#organization"
    },
    "breadcrumb": {
      "@id": "https://uncodemy.com/python-data-cleaning-pdf#breadcrumb"
    },
    "about": {
      "@type": "Thing",
      "name": "Python Data Cleaning"
    },
    "primaryImageOfPage": {
      "@type": "ImageObject",
      "url": "https://uncodemy.com/image/logo-og.jpg"
    },
    "dateModified": "2026-10-26"
  },
  {
    "@type": "FAQPage",
    "@id": "https://uncodemy.com/python-data-cleaning-pdf#faq",
    "inLanguage": "en-IN",
    "isPartOf": {
      "@id": "https://uncodemy.com/python-data-cleaning-pdf#webpage"
    },
    "mainEntity": [
      {
        "@type": "Question",
        "name": "How do I clean data in Python?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "Use pandas to standardise text, remove duplicates, convert types and dates, handle missing values and outliers with documented rules, and validate the result with checks."
        }
      },
      {
        "@type": "Question",
        "name": "How do I handle missing values in pandas?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "Detect them with isna, then drop, fill with a justified value or flag them, depending on the column's meaning. Record what you did."
        }
      },
      {
        "@type": "Question",
        "name": "How do I convert text dates in pandas?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "Use pd.to_datetime, with errors set to coerce to expose unparseable values, and check the day-month order carefully."
        }
      },
      {
        "@type": "Question",
        "name": "How do I make cleaning repeatable?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "Put the rules in a function or script, add assertions for key checks and keep it under version control."
        }
      },
      {
        "@type": "Question",
        "name": "Should I delete outliers?",
        "acceptedAnswer": {
          "@type": "Answer",
          "text": "Not automatically. Investigate whether they are errors or real events and treat them accordingly."
        }
      }
    ]
  }
]
}
</script>

<!-- ADD AFTER THE PDF TITLES ARE FINAL (remove this wrapper, fill every [..] value):
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ItemList",
  "@id": "https://uncodemy.com/python-data-cleaning-pdf#pdfs",
  "name": "Python Data Cleaning PDF downloads",
  "numberOfItems": 3,
  "itemListElement": [

```

```

{
  "@type": "ListItem",
  "position": 1,
  "item": {
    "@type": "DigitalDocument",
    "name": "[PDF 1 TITLE]",
    "description": "[PDF 1 ONE-LINE DESCRIPTION]",
    "encodingFormat": "application/pdf",
    "url": "https://uncodemy.com/pdf/python-data-cleaning.pdf",
    "inLanguage": "en",
    "publisher": {
      "@id": "https://uncodemy.com/#organization"
    }
  }
},
{
  "@type": "ListItem",
  "position": 2,
  "item": {
    "@type": "DigitalDocument",
    "name": "[PDF 2 TITLE]",
    "description": "[PDF 2 ONE-LINE DESCRIPTION]",
    "encodingFormat": "application/pdf",
    "url": "https://uncodemy.com/pdf/python-data-cleaning-2.pdf",
    "inLanguage": "en",
    "publisher": {
      "@id": "https://uncodemy.com/#organization"
    }
  }
},
{
  "@type": "ListItem",
  "position": 3,
  "item": {
    "@type": "DigitalDocument",
    "name": "[PDF 3 TITLE]",
    "description": "[PDF 3 ONE-LINE DESCRIPTION]",
    "encodingFormat": "application/pdf",
    "url": "https://uncodemy.com/pdf/python-data-cleaning-3.pdf",
    "inLanguage": "en",
    "publisher": {
      "@id": "https://uncodemy.com/#organization"
    }
  }
}
}
}
</script>
-->

```

DEVELOPER CHECKLIST

- ☐ Final URL live at `https://uncodemy.com/python-data-cleaning.pdf`; canonical, og:url and sitemap identical.
- ☐ Each button: `<a href="/pdf/python-data-cleaning.pdf" download type="application/pdf">`, its own file, size shown.
- ☐ If a form precedes download: consent checkbox, Privacy Policy link, nothing pre-ticked.
- ☐ One `<h1>`; real HTML tables; code samples in `<pre><code>`.
- ☐ Remove any `<meta name="keywords">` from the template.
- ☐ Test in Google Rich Results Test and validator.schema.org after publishing.
- ☐ Add to sitemap and any PDF/resources hub.

STILL NEEDED FROM YOU

#	Item
1	Course PDF title, description, page count, size, and the file URL for each page
2	Free or gated (and form fields if gated)
3	Author, reviewer, publish and last-reviewed dates
4	Course URL for the CTA (lines marked <code>[CONFIRM]</code>)
5	Feedback email
6	Confirmation that Uncodemy owns the rights to every PDF