

# Data challenges

---

- Non-standardized data
- Inconveniently structured data
  - Tidy-ing data
  - Data with multiple factors
- Duplicate data
- Incorrect values
- Missing values



# Standardizing data

Raw Year	Standardized
2019	2019
'19	2019

Raw Medication	Standardized
azithromycin	azithromycin
Zithromax	azithromycin

Raw Name	Standardized
McDougal	McDougal
mcdougal	McDougal

Raw Unit	Standardized
micron	µm
µm	µm

myocardial infarction[MeSH Ter X

https://pubmed.ncbi.nlm.nih.gov/?term=myocardial+infarcti ...

NIH National Library of Medicine  
National Center for Biotechnology Information

Log in

PubMed.gov

myocardial infarction[MeSH Terms]

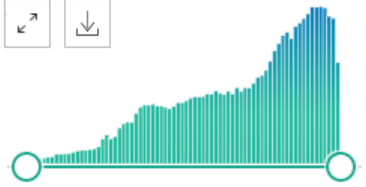
Search

Advanced Create alert Create RSS User Guide

Save Email Send to Sorted by: Best match Display options

MY NCBI FILTERS

RESULTS BY YEAR



1945 2020

TEXT AVAILABILITY

☐ Abstract

☐ Free full text

☐ Full text

ARTICLE ATTRIBUTE

☐ Associated data

174,312 results

☐ **Myocardial Infarction: Symptoms and Treatments.**

1 Lu L, Liu M, Sun R, Zheng Y, Zhang P.  
Cell Biochem Biophys. 2015 Jul;72(3):865-7. doi: 10.1007/s12013-015-0553-4.  
PMID: 25638347 Review.

“ Cite ↗ Share

☐ **The acute myocardial infarction.**

2 Pollard TJ.  
Prim Care. 2000 Sep;27(3):631-49;vi. doi: 10.1016/s0095-4543(05)70167-6.  
PMID: 10918673 Review.

“ Cite ↗ Share

☐ **Acute Complications of Myocardial Infarction in the Current Era: Diagnosis and Management.**

3 Bajaj A, Sethi A, Rathor P, Suppogu N, Sethi A.  
J Investig Med. 2015 Oct;63(7):844-55. doi: 10.1097/JIM.0000000000000232.

Feedback

myocardial infarction[MeSH Terms] X

https://pubmed.ncbi.nlm.nih.gov/?term=myocardial+infarctio ...

NIH National Library of Medicine  
National Center for Biotechnology Information

Log in

PubMed.gov

myocardial infarction[MeSH Terms]

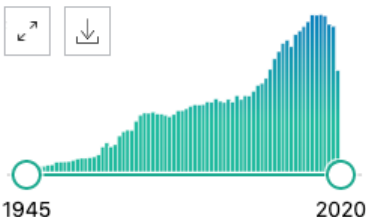
Advanced Create alert Create RSS

Save Email Send to

MY NCBI FILTERS

174,312 results

RESULTS BY YEAR



1945 2020

TEXT AVAILABILITY

☐ Abstract

☐ Free full text

☐ Full text

ARTICLE ATTRIBUTE

☐ Associated data

☐ Myocardial Infarction: Symptoms and Treatment.

1 Lu L, Liu M, Sun R, Zheng Y, Zhang P.  
Cell Biochem Biophys. 2015 Jul;72(3):865-7. doi: 10.1007/s12013-015-0400-0.  
PMID: 25638347 Review.

“ Cite Share

☐ The acute myocardial infarction.

2 Pollard TJ.  
Prim Care. 2000 Sep;27(3):631-49;vi. doi: 10.1016/S0883-5963(00)00000-0.  
PMID: 10918673 Review.

“ Cite Share

☐ Acute Complications of Myocardial Infarction: Management.

3 Bajaj A, Sethi A, Rathor P, Suppogu N, Sethi A.  
J Investig Med. 2015 Oct;63(7):844-55. doi: 10.1093/jimab/63.7.844.

## Publication types

> Review

## MeSH terms

> Humans

> Myocardial Infarction / diagnosis\*

> Myocardial Infarction / etiology

> Myocardial Infarction / prevention & control

> Myocardial Infarction / therapy

## LinkOut – more resources

### Full Text Sources

Springer

### Medical

MedlinePlus Health Information

### Miscellaneous

NCI CPTAC Assay Portal

MeSH Browser

https://meshb.nlm.nih.gov/record/ui?name=Myocardial

Search

# Myocardial Infarction MeSH Descriptor Data 2020

Details Qualifiers MeSH Tree Structures Concepts

<b>MeSH Heading</b>	Myocardial Infarction
<b>Tree Number(s)</b>	C14.280.647.500 C14.907.585.500 C23.550.513.355.750 C23.550.717.489.750
<b>Unique ID</b>	D009203
<b>RDF Unique Identifier</b>	<a href="http://id.nlm.nih.gov/mesh/D009203">http://id.nlm.nih.gov/mesh/D009203</a>
<b>Annotation</b>	do not coordinate with <a href="#">ACUTE DISEASE</a> for "acute infarct"
<b>Scope Note</b>	<a href="#">NECROSIS</a> of the <a href="#">MYOCARDIUM</a> caused by an obstruction of the blood supply to the heart ( <a href="#">CORONARY CIRCULATION</a> ).
<b>Entry Term(s)</b>	Cardiovascular Stroke Heart Attack Myocardial Infarct
<b>NLM Classification #</b>	WG 310
<b>See Also</b>	<a href="#">Heart Rupture, Post-Infarction</a>
<b>Public MeSH Note</b>	79; was MYOCARDIAL INFARCT 1963-78
<b>Online Note</b>	use MYOCARDIAL INFARCTION to search MYOCARDIAL INFARCT 1966-78
<b>History Note</b>	79; was MYOCARDIAL INFARCT 1963-78
<b>Date Established</b>	1966/01/01
<b>Date of Entry</b>	1999/01/01
<b>Revision Date</b>	2019/07/01

page delivered in 0.146s

Copyright , Privacy , Accessibility , Site Map , Viewers and Players

U.S. National Library of Medicine, 8600 Rockville Pike, Bethesda, MD

USA.gov

MeSH Browser

https://meshb.nlm.nih.gov/record/ui?name=Myocardial

Search

NIH U.S. National Library of Medicine

MeSH

Search Tree View MeSH on Demand MeSH 2019 MeSH Suggestions About MeSH Browser Contact Us

# Myocardial Infarction MeSH Descriptor Data 2020

Details Qualifiers MeSH Tree Structures Concepts

- Cardiovascular Diseases [C14]
  - Heart Diseases [C14.280]
    - Myocardial Ischemia [C14.280.647]
      - Acute Coronary Syndrome [C14.280.647.124]
      - Angina Pectoris [C14.280.647.187] +
      - Coronary Disease [C14.280.647.250] +
      - Kounis Syndrome [C14.280.647.375]
      - Myocardial Infarction [C14.280.647.500] -**
        - Anterior Wall Myocardial Infarction [C14.280.647.500.093]
        - Inferior Wall Myocardial Infarction [C14.280.647.500.187]
        - Non-ST Elevated Myocardial Infarction [C14.280.647.500.469]
        - Shock, Cardiogenic [C14.280.647.500.750]
        - ST Elevation Myocardial Infarction [C14.280.647.500.875]
      - Myocardial Reperfusion Injury [C14.280.647.625]
- Pathological Conditions, Signs and Symptoms [C23]
  - Pathologic Processes [C23.550]
    - Necrosis [C23.550.717]
      - Infarction [C23.550.717.489]
        - Brain Infarction [C23.550.717.489.250] +
        - Hepatic Infarction [C23.550.717.489.500]
        - Myocardial Infarction [C23.550.717.489.750] -

# Some Ontologies and Terminologies

- MeSH
  - Medical Subject Headings.
- RxNorm
  - All medications available in the United States.
- UMLS
  - Unified Medical Language System. (Use requires a free license and annual reporting.)
- GO
  - Gene Ontology.
- SNOMED-CT
  - Clinical terms.
- ChEBI
  - Chemical Entities of Biological Interest.

# On dates

- ISO 8601 is an international standard for date-time information.
  - `20191001T182618+0000`
- A challenge with this is that it requires dates be parsed again to do calculations.
  - It may be easier to store this as separate fields: year, month, day, hour, minute, seconds.
- Can use `pd.to_datetime` to convert to `DateTime` objects to allow subtraction, comparison.
- An alternative: Unix time
  - Number of seconds since 1 January 1970 UTC.
  - Returned by `time.time()` or e.g.

```
((pd.to_datetime('June 10, 2020') -  
  pd.to_datetime('January 1, 1970')) /  
  pd.Timedelta('1 sec'))
```





# Tidy Data

A data structuring approach.

Every variable has its own column.

Every observation has its own row.

Every value has its own cell.

Date	fernando_height	samantha_height	raul_height	pi_height
1990-01-01	0.50	0.65	1.70	1.4
1991-01-01	0.75	0.87	1.78	1.44
1992-01-01	0.87	0.94	1.84	1.49
1993-01-01	0.96	1.01	1.87	1.57
1994-01-01	1.03	1.08	1.89	1.64

Melting

Date	fernando_height	samantha_height	raul_height	pi_height
1990-01-01	0.50	0.65	1.70	1.4
1991-01-01	0.75	0.87	1.78	1.44
1992-01-01	0.87	0.94	1.84	1.49
1993-01-01	0.96	1.01	1.87	1.57
1994-01-01	1.03	1.08	1.89	1.64



```
pd.melt(heights,
        id_vars=['Date'],
        value_vars=['fernando_height',
                    'samantha_height',
                    'raul_height',
                    'pi_height'])
```

	Date	variable	value
0	1990-01-01	fernando_height	0.50
1	1991-01-01	fernando_height	0.75
2	1992-01-01	fernando_height	0.87
3	1993-01-01	fernando_height	0.96
4	1994-01-01	fernando_height	1.03
5	1990-01-01	samantha_height	0.65
6	1991-01-01	samantha_height	0.87
7	1992-01-01	samantha_height	0.94
8	1993-01-01	samantha_height	1.01
⋮	⋮	⋮	⋮

# Melting

Date	fernando_height	samantha_height	raul_height	pi_height
1990-01-01	0.50	0.65	1.70	1.4
1991-01-01	0.75	0.87	1.78	1.44
1992-01-01	0.87	0.94	1.84	1.49
1993-01-01	0.96	1.01	1.87	1.57
1994-01-01	1.03	1.08	1.89	1.64



```
pd.melt(heights,
        id_vars=['Date'],
        value_vars=['fernando_height',
                    'samantha_height',
                    'raul_height',
                    'pi_height'],
        var_name='name',
        value_name='height'
    )
```

	Date	name	height
0	1990-01-01	fernando_height	0.50
1	1991-01-01	fernando_height	0.75
2	1992-01-01	fernando_height	0.87
3	1993-01-01	fernando_height	0.96
4	1994-01-01	fernando_height	1.03
5	1990-01-01	samantha_height	0.65
6	1991-01-01	samantha_height	0.87
7	1992-01-01	samantha_height	0.94
8	1993-01-01	samantha_height	1.01
:	:	:	:

# Melting

# Data with multiple factors



“8-mg Zofran”

Dosage (mg): 8

Medicine: ondansetron



“effusion of the right  
knee”

Condition: “knee effusion”

Side: “right”



“white male”

Gender: male

Race: white

# Duplicate data

- Sometimes a data point (row) may be listed more than once, especially if manual entry was involved.
- But be careful: depending on how your data is structured, it may also be the case that data should appear more than once.
  - Imagine, e.g. a patient sent home from the hospital only to return later that day with the same conditions.

# Duplicate data example

```
import pandas as pd
```

```
patients = [  
    (1002, 'Smith', 'John', 42, '20191001', 'diabetes'),  
    (4261, 'Smith', 'Jane', 46, '20190510', 'pulmonary embolism'),  
    (1002, 'Smith', 'John', 42, '20191001', 'diabetes'),  
    (4171, 'Smith', 'Janet', 16, '20190909', 'acne')  
]
```

```
data = pd.DataFrame(  
    patients,  
    columns=['pid', 'last', 'first', 'age', 'date', 'condition'])
```

# Duplicate data example

- See duplicate rows:

```
>>> data[data.duplicated()]
```

	pid	last	first	age	date	condition
2	1002	Smith	John	42	20191001	diabetes



# Duplicate data example

Pid should uniquely identify a patient.

Date and pid *almost* uniquely identifies an encounter.

- See duplicate rows:

```
>>> data[data.duplicated()]
```

	pid	last	first	age	date	condition
2	1002	Smith	John	42	20191001	diabetes



# Duplicate data example

- See duplicate rows:

```
>>> data[data.duplicated()]
```

	pid	last	first	age	date	condition
2	1002	Smith	John	42	20191001	diabetes

- Drop duplicate rows

```
>>> deduplicated_data = data.drop_duplicates()
```

```
>>> deduplicated_data
```

	pid	last	first	age	date	condition
0	1002	Smith	John	42	20191001	diabetes
1	4261	Smith	Jane	46	20190510	pulmonary embolism
3	4171	Smith	Janet	16	20190909	acne

# Duplicate data example

- Both `data.duplicated` and `data.drop_duplicates` take an optional *subset* keyword argument specifying which columns to pay attention to.

```
>>> data.duplicated(subset=['last'])
0      False
1       True
2       True
3       True
dtype: bool
```

- Define and check ranges
  - If a person is 57 years old, that is plausible. If a person is 577 years old, then maybe there is something wrong.
- Check categorical values
  - e.g. is the "State" field correct? We know the list of all possible states.
- Look for inconsistencies
  - e.g. City: "New Haven", Zip: "90210"
- Look at outliers
  - If only one person has a disease, it could be very rare... or it could be a typo.
- Validate when possible.

Incorrect values