Chapter 4 - Ex2: Tidying Data

Câu 1: Pew Research Center

Cho dữ liệu: pew-raw.csv. Bộ dữ liệu này khám phá mối quan hệ giữa thu nhập và tôn giáo (income & religion)

- Đọc dữ liệu
- Xem xét vấn đề về dữ liệu cần khắc phục
- Chuẩn lại dữ liệu để khắc phục vấn dề trên

Câu 2: Billboard Top 100

Cho dữ liệu billboard.csv. Bộ dữ liệu này đại diện cho thứ hạng hàng tuần của các bài hát kể từ thời điểm chúng lọt vào Top 100 của Billboard cho đến 75 tuần tiếp theo.

- Đọc dữ liệu
- Xem xét vấn đề về dữ liệu cần khắc phục
- Chuẩn lại dữ liệu để khắc phục vấn dề trên
- Sau khi chuẩn lại dữ liệu, có điều gì phát sinh cần khắc phục tiếp theo không? Nếu có thì hãy chuẩn lại dữ liệu mới có ở câu trên

Câu 3: Tubercolosis

Cho dữ liệu tb-raw.csv . Bộ dữ liệu này ghi lại số ca bệnh lao được xác nhận theo quốc gia, năm, tuổi và giới tính.

- Đọc dữ liệu
- Xem xét vấn đề về dữ liệu cần khắc phục
- Chuẩn lại dữ liệu để khắc phục vấn dề trên

Câu 4: Global Historical Climatology Network

Cho dữ liệu weather-raw.csv. Bộ dữ liệu này đại diện cho các bản ghi thời tiết hàng ngày cho một trạm thời tiết (MX17004) ở Mexico trong 5 tháng năm 2010.

- Đọc dữ liệu
- Xem xét vấn đề về dữ liệu cần khắc phục
- Chuẩn lại dữ liệu để khắc phục vấn dề trên

Câu 1: Gợi ý

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv("pew-raw.csv")
df
```

Out[2]:

	religion	<$10k	$10-20k	$20-30k	$30-40k	$40-50k	$50-75k
0	Agnostic	27	34	60	81	76	137
1	Atheist	12	27	37	52	35	70
2	Buddhist	27	21	30	34	33	58
3	Catholic	418	617	732	670	638	1116
4	Dont know/refused	15	14	15	11	10	35
5	Evangelical Prot	575	869	1064	982	881	1486
6	Hindu	1	9	7	9	11	34
7	Historically Black Prot	228	244	236	238	197	223
8	Jehovahs Witness	20	27	24	24	21	30
9	Jewish	19	19	25	25	30	95

Vấn đề cần khắc phục:

- Tên cột chứa giá trị thay vì chứa biến, trong từng cell chứa tần suất

In [3]:

```
df_after = pd.melt(frame=df,
                   id_vars = ["religion"],
                   var_name="income",
                   value_name="freq")
#df_after = df_after.sort_values(by=["religion"])
df_after.head(10)
```

Out[3]:

	religion	income	freq
0	Agnostic	<$10k	27
1	Atheist	<$10k	12
2	Buddhist	<$10k	27
3	Catholic	<$10k	418
4	Dont know/refused	<$10k	15
5	Evangelical Prot	<$10k	575
6	Hindu	<$10k	1
7	Historically Black Prot	<$10k	228
8	Jehovahs Witness	<$10k	20
9	Jewish	<$10k	19

Câu 2: Gợi ý

In [4]:

```
df1 = pd.read_csv("billboard.csv", encoding="mac_latin2")
df1.head()
```

Out[4]:

	year	artist.inverted	track	time	genre	date.entered	date.peaked	x1st.week	x2nd.
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	2000-09-23	2000-11-18	78	
1	2000	Santana	Maria, Maria	4:18	Rock	2000-02-12	2000-04-08	15	
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	1999-10-23	2000-01-29	71	
3	2000	Madonna	Music	3:45	Rock	2000-08-12	2000-09-16	41	
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	2000-08-05	2000-10-14	57	

5 rows × 83 columns

Vấn đề cần khắc phục:

- Tiêu đề cột chứa giá trị thay vì chứa biến, trong từng cell chứa thứ hạng
- Nếu một bài hát nằm trong Top 100 dưới 75 tuần, các cột còn lại chứa đầy các giá trị bị thiếu (NaN)

In [5]:

```
df1_after = pd.melt(frame=df1,
                    id_vars = ["year", "artist.inverted", "track",
                               "time", "genre", "date.entered",
                               "date.peaked"],
                    var_name="week", value_name="rank")

df1_after.head(10)
```

Out[5]:

	year	artist.inverted	track	time	genre	date.entered	date.peaked	week	rank
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	2000-09-23	2000-11-18	x1st.week	78.0
1	2000	Santana	Maria, Maria	4:18	Rock	2000-02-12	2000-04-08	x1st.week	15.0
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	1999-10-23	2000-01-29	x1st.week	71.0
3	2000	Madonna	Music	3:45	Rock	2000-08-12	2000-09-16	x1st.week	41.0
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	2000-08-05	2000-10-14	x1st.week	57.0
5	2000	Janet	Doesn't Really Matter	4:17	Rock	2000-06-17	2000-08-26	x1st.week	59.0
6	2000	Destiny's Child	Say My Name	4:31	Rock	1999-12-25	2000-03-18	x1st.week	83.0
7	2000	Iglesias, Enrique	Be With You	3:36	Latin	2000-04-01	2000-06-24	x1st.week	63.0
8	2000	Sisqo	Incomplete	3:52	Rock	2000-06-24	2000-08-12	x1st.week	77.0
9	2000	Lonestar	Amazed	4:25	Country	1999-06-05	2000-03-04	x1st.week	81.0

In [6]:

```
# Chuẩn hóa dữ liệu
#__ Thay chuỗi trong tuần bằng số
#__ Xóa bỏ các dòng dữ liệu chứa NaN
#__ Xem kiểu của rank, nếu chưa là số thì chuyển thành số
```

In [7]:

```
df1_after["week"] = df1_after['week'].str.extract('(\d+)',
                                                  expand=False).astype(int)
```

In [8]:

```
# Cleaning out unnecessary rows
df1_after = df1_after.dropna()
```

In [9]:

```
df1_after["rank"] = df1_after["rank"].astype(int)
```

In [10]:

```
df1_after.head()
```

Out[10]:

	year	artist.inverted	track	time	genre	date.entered	date.peaked	week	rank
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	2000-09-23	2000-11-18	1	78
1	2000	Santana	Maria, Maria	4:18	Rock	2000-02-12	2000-04-08	1	15
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	1999-10-23	2000-01-29	1	71
3	2000	Madonna	Music	3:45	Rock	2000-08-12	2000-09-16	1	41
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	2000-08-05	2000-10-14	1	57

Vấn đề mới phát sinh:

- Nhiều đơn vị quan sát (song & rank của nó) trong một bảng duy nhất.

In [11]:

```
# Tạo một dataframe songs và các thông tin của nó
# Tạo một dataframe ranks chứa id của song và week kèm theo rank
```

In [12]:

```
songs_cols = ["year", "artist.inverted", "track", "time", "genre"]
songs = df1_after[songs_cols].drop_duplicates()
songs = songs.reset_index(drop=True)
songs["song_id"] = songs.index
songs.head(10)
```

Out[12]:

	year	artist.inverted	track	time	genre	song_id
0	2000	Destiny's Child	Independent Women Part I	3:38	Rock	0
1	2000	Santana	Maria, Maria	4:18	Rock	1
2	2000	Savage Garden	I Knew I Loved You	4:07	Rock	2
3	2000	Madonna	Music	3:45	Rock	3
4	2000	Aguilera, Christina	Come On Over Baby (All I Want Is You)	3:38	Rock	4
5	2000	Janet	Doesn't Really Matter	4:17	Rock	5
6	2000	Destiny's Child	Say My Name	4:31	Rock	6
7	2000	Iglesias, Enrique	Be With You	3:36	Latin	7
8	2000	Sisqo	Incomplete	3:52	Rock	8
9	2000	Lonestar	Amazed	4:25	Country	9

In [13]:

```
ranks = pd.merge(df1_after, songs, on=["year","artist.inverted",
                                       "track", "time", "genre"])
ranks = ranks[["song_id", "date.entered",
               "date.peaked", "week","rank"]]
ranks.head(10)
```

Out[13]:

	song_id	date.entered	date.peaked	week	rank
0	0	2000-09-23	2000-11-18	1	78
1	0	2000-09-23	2000-11-18	2	63
2	0	2000-09-23	2000-11-18	3	49
3	0	2000-09-23	2000-11-18	4	33
4	0	2000-09-23	2000-11-18	5	23
5	0	2000-09-23	2000-11-18	6	15
6	0	2000-09-23	2000-11-18	7	7
7	0	2000-09-23	2000-11-18	8	5
8	0	2000-09-23	2000-11-18	9	1
9	0	2000-09-23	2000-11-18	10	1

In [14]:

```
# Tạo cột date
ranks['date'] = pd.to_datetime(ranks['date.entered']) + \
    pd.to_timedelta(ranks['week'], unit='w') - pd.DateOffset(weeks=1)
```

In [15]:

```
ranks.head()
```

Out[15]:

	song_id	date.entered	date.peaked	week	rank	date
0	0	2000-09-23	2000-11-18	1	78	2000-09-23
1	0	2000-09-23	2000-11-18	2	63	2000-09-30
2	0	2000-09-23	2000-11-18	3	49	2000-10-07
3	0	2000-09-23	2000-11-18	4	33	2000-10-14
4	0	2000-09-23	2000-11-18	5	23	2000-10-21

In [16]:

```
ranks = ranks.drop(["date.entered", "date.peaked"], axis=1)
```

In [17]:

```
ranks.head()
```

Out[17]:

	song_id	week	rank	date
0	0	1	78	2000-09-23
1	0	2	63	2000-09-30
2	0	3	49	2000-10-07
3	0	4	33	2000-10-14
4	0	5	23	2000-10-21

Câu 3: Gợi ý

In [18]:

```
df2 = pd.read_csv("tb-raw.csv")
df2
```

Out[18]:

	country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
0	AD	2000	0.0	0.0	1.0	0.0	0	0	0.0	NaN	NaN
1	AE	2000	2.0	4.0	4.0	6.0	5	12	10.0	NaN	3.0
2	AF	2000	52.0	228.0	183.0	149.0	129	94	80.0	NaN	93.0
3	AG	2000	0.0	0.0	0.0	0.0	0	0	1.0	NaN	1.0
4	AL	2000	2.0	19.0	21.0	14.0	24	19	16.0	NaN	3.0
5	AM	2000	2.0	152.0	130.0	131.0	63	26	21.0	NaN	1.0
6	AN	2000	0.0	0.0	1.0	2.0	0	0	0.0	NaN	0.0
7	AO	2000	186.0	999.0	1003.0	912.0	482	312	194.0	NaN	247.0
8	AR	2000	97.0	278.0	594.0	402.0	419	368	330.0	NaN	121.0
9	AS	2000	NaN	NaN	NaN	NaN	1	1	NaN	NaN	NaN

Vấn đề cần khắc phục:

- Tên cột chứa giá trị thay vì chứa biến, ngoài ra nó còn chứa 2 thông tin là giới tính và độ tuổi
- Trong cell chứa hỗn hợp giá trị (value, 0 và NaN)

In [19]:

```
# Chuyển dữ liệu theo định dạng country, year, và sex_and_age
df2_after = pd.melt(df2, id_vars=["country","year"],
                    value_name="cases", var_name="sex_and_age")
df2_after.head()
```

Out[19]:

	country	year	sex_and_age	cases
0	AD	2000	m014	0.0
1	AE	2000	m014	2.0
2	AF	2000	m014	52.0
3	AG	2000	m014	0.0
4	AL	2000	m014	2.0

In [20]:

```
# Tách sex_and_age thành 2 cột: sex và age
# Extract Sex, Age lower bound and Age upper bound group
tmp_df = df2_after["sex_and_age"].str.extract("(\D)(\d+)(\d{2})")

# Name columns
tmp_df.columns = ["sex", "age_lower", "age_upper"]

# Create `age`column based on `age_lower` and `age_upper`
tmp_df["age"] = tmp_df["age_lower"] + "-" + tmp_df["age_upper"]
```

In [21]:

```
# Merge
df2_after = pd.concat([df2_after, tmp_df], axis=1)
df2_after.head()
```

Out[21]:

	country	year	sex_and_age	cases	sex	age_lower	age_upper	age
0	AD	2000	m014	0.0	m	0	14	0-14
1	AE	2000	m014	2.0	m	0	14	0-14
2	AF	2000	m014	52.0	m	0	14	0-14
3	AG	2000	m014	0.0	m	0	14	0-14
4	AL	2000	m014	2.0	m	0	14	0-14

Câu 4: Gợi ý

In [22]:

```
df3 = pd.read_csv("weather-raw.csv")
df3
```

Out[22]:

	id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
0	MX17004	2010	1	tmax	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	MX17004	2010	1	tmin	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	MX17004	2010	2	tmax	NaN	27.3	24.1	NaN	NaN	NaN	NaN	NaN
3	MX17004	2010	2	tmin	NaN	14.4	14.4	NaN	NaN	NaN	NaN	NaN
4	MX17004	2010	3	tmax	NaN	NaN	NaN	NaN	32.1	NaN	NaN	NaN
5	MX17004	2010	3	tmin	NaN	NaN	NaN	NaN	14.2	NaN	NaN	NaN
6	MX17004	2010	4	tmax	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	MX17004	2010	4	tmin	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	MX17004	2010	5	tmax	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	MX17004	2010	5	tmin	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Vấn đề cần khắc phục

- Các biến được lưu trữ trong cả các dòng (tmin, tmax) và các cột (days).
- Tên cột chứa giá trị day thay vì chứa biến
- Trong từng cell chứa temperature nhưng có rất nhiều giá trị NaN
  => Để làm cho tập dữ liệu này gọn gàng => di chuyển ba biến bị đặt sai (tmin, tmax và day) thành ba cột riêng lẻ: tmin. tmax và date.

In [23]:

```
df3_after = pd.melt(df3, id_vars=["id", "year","month","element"],
                    var_name="day_raw")
df3_after.head(10)
```

Out[23]:

	id	year	month	element	day_raw	value
0	MX17004	2010	1	tmax	d1	NaN
1	MX17004	2010	1	tmin	d1	NaN
2	MX17004	2010	2	tmax	d1	NaN
3	MX17004	2010	2	tmin	d1	NaN
4	MX17004	2010	3	tmax	d1	NaN
5	MX17004	2010	3	tmin	d1	NaN
6	MX17004	2010	4	tmax	d1	NaN
7	MX17004	2010	4	tmin	d1	NaN
8	MX17004	2010	5	tmax	d1	NaN
9	MX17004	2010	5	tmin	d1	NaN

In [24]:

```
# Tạo cột day chỉ chứa ngày
df3_after["day"] = df3_after["day_raw"].str.extract("d(\d+)",
                                                    expand=False)
```

In [25]:

```
df3_after.head()
```

Out[25]:

	id	year	month	element	day_raw	value	day
0	MX17004	2010	1	tmax	d1	NaN	1
1	MX17004	2010	1	tmin	d1	NaN	1
2	MX17004	2010	2	tmax	d1	NaN	1
3	MX17004	2010	2	tmin	d1	NaN	1
4	MX17004	2010	3	tmax	d1	NaN	1

In [26]:

```
# Chuyển dữ liệu thành số
df3_after[["year","month","day"]] = \
        df3_after[["year", "month","day"]].apply(lambda x: \
                                                 pd.to_numeric(x,
                                                               errors='ignore'))
```

In [27]:

```
df3_after.head()
```

Out[27]:

	id	year	month	element	day_raw	value	day
0	MX17004	2010	1	tmax	d1	NaN	1
1	MX17004	2010	1	tmin	d1	NaN	1
2	MX17004	2010	2	tmax	d1	NaN	1
3	MX17004	2010	2	tmin	d1	NaN	1
4	MX17004	2010	3	tmax	d1	NaN	1

In [28]:

```
# Tạo cột date chứa cả day, month, year
import datetime
def create_date_from_year_month_day(row):
    return datetime.datetime(year=row["year"],
                             month=int(row["month"]), day=row["day"])
```

In [29]:

```
df3_after["date"] = df3_after.apply(lambda row: \
                                    create_date_from_year_month_day(row),
                                                        axis=1)
```

In [30]:

```
# bỏ các cột thừa
df3_after = df3_after.drop(['year',"month","day", "day_raw"], axis=1)
```

In [31]:

```
# Bỏ các dòng dữ liệu NaN
df3_after = df3_after.dropna()
```

In [32]:

```
df3_after.head()
```

Out[32]:

	id	element	value	date
12	MX17004	tmax	27.3	2010-02-02
13	MX17004	tmin	14.4	2010-02-02
22	MX17004	tmax	24.1	2010-02-03
23	MX17004	tmin	14.4	2010-02-03
44	MX17004	tmax	32.1	2010-03-05

In [ ]: