

ます。

判定した結果、数値として取り込まれている対象をflg_is_serialに格納します。

2行目は、後の検証のために、対象件数を表示しています。今回は22件が数値として取り込まれている事が分かります。

それでは、数値で取り込まれている登録日を補正していきましょう。

```
fromSerial = pd.to_timedelta(kokyaku_data.loc[flg_is_serial, "登録日"].astype("float"), unit="D") + pd.to_datetime("1900/01/01")
fromSerial
```

■図2-19：数値から日付に変換

```
In [19]: fromSerial = pd.to_timedelta(kokyaku_data.loc[flg_is_serial, "登録日"].astype("float"), unit="D") + pd.to_datetime("1900/01/01")
fromSerial
Out[19]: 1    2017-02-18
          3    2017-05-19
          4    2018-01-29
          21   2017-07-06
          27   2017-06-17
          47   2017-01-08
          49   2017-07-15
          53   2017-04-10
          76   2018-03-31
          80   2018-01-12
          99   2017-06-01
          114   2018-06-05
          118   2018-01-31
          122   2018-04-18
          139   2017-05-27
          143   2017-03-26
          155   2017-01-21
          172   2018-03-24
          179   2017-01-10
          183   2017-07-26
          186   2018-07-15
          192   2018-06-10
          Name: 登録日, dtype: datetime64[ns]
```

この処理では、pd.to_timedelta()関数を用いて、数値(シリアル値)から日付に変換しています。対象のデータは.locを用いflg_is_serialの条件でデータを抽出します。

変換した結果を表示しています。

Excelの該当データと比較すると、正しくデータが変換されている事が確認できます。

次に、日付として取り込まれているデータも、書式統一のために処理します。

```
fromString = pd.to_datetime(kokyaku_data.loc[~flg_is_serial, "登録日"])
fromString
```

■図2-20：日付として取り込まれている対象の書式変更結果

```
In [20]: fromString = pd.to_datetime(kokyaku_data.loc[~flg_is_serial, "登録日"])
fromString
Out[20]: 0    2018-01-04
          2    2018-01-07
          5    2017-06-20
          6    2018-06-11
          7    2017-05-19
          8    2018-02-12
          9    2017-07-05
          10   2018-03-31
          11   2017-04-22
          12   2018-03-09
          13   2017-03-13
          14   2018-01-24
          15   2017-06-09
          16   2018-05-02
          17   2017-02-05
          18   2018-07-10
          19   2017-07-08
          20   2018-03-12
          22   2018-02-28
          23   2017-05-10
```

2018/01/04とスラッシュ区切りの書式を、ハイフン区切りに統一するために処理を行いました。

数値から日付に補正したデータと、書式を変更したデータを結合しデータを更新します。

```
kokyaku_data["登録日"] = pd.concat([fromSerial, fromString])
kokyaku_data
```

■図2-21：日付更新結果

```
In [21]: kokyaku_data["登録日"] = pd.concat([fromSerial, fromString])
kokyaku_data
Out[21]:
```

	顧客名	かな	地域	メールアドレス	登録日
0	渡辺ひとみ	すがひとみ	H市	tsuga_hitori@example.com	2018-01-04
1	岡田航也	おかた としや	E市	okada_toshiya@example.com	2017-02-18
2	斎藤麻	はが のぞみ	A市	haga_nozomi@example.com	2018-01-07
3	荻野愛	おぎの あい	F市	ogino_ai@example.com	2017-05-19
4	栗田幸一	くりた けんいち	E市	kurita_kenichi@example.com	2018-01-29
5	梅沢麻理	うめざわ まお	A市	umezawa_mao@example.com	2017-06-20
6	相澤ひとり	あいはら ひとり	H市	aihara_hitori@example.com	2018-06-11
7	新村文貴	にいむら ぶんき	B市	nimura_bunshi@example.com	2017-05-19
8	石川まさみ	いしかわ まさみ	G市	ishikawa_masami@example.com	2018-02-12
9	山崎正樹	やまざき まさき	C市	yamazaki_masaki@example.com	2017-07-05