**Desafio: normalizando o JSON obtido na API**

Chegou a hora de testar os conhecimentos desenvolvidos durante a aula.

O DataFrame mostrado abaixo foi gerado após a obtenção de dados da API [JSONPlaceholder](https://jsonplaceholder.typicode.com/users" \t "_blank):

| **index** | **id** | **name** | **username** | **email** | **address** | **phone** | **website** | **company** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | Leanne Graham | Bret | Sincere@april.biz | {'street': 'Kulas Light', 'suite': 'Apt. 556', 'city': 'Gwenborough', 'zipcode': '92998-3874', 'geo': {'lat': '-37.3159', 'lng': '81.1496'}} | 1-770-736-8031 x56442 | hildegard.org | {'name': 'Romaguera-Crona', 'catchPhrase': 'Multi-layered client-server neural-net', 'bs': 'harness real-time e-markets'} |
| 1 | 2 | Ervin Howell | Antonette | Shanna@melissa.tv | {'street': 'Victor Plains', 'suite': 'Suite 879', 'city': 'Wisokyburgh', 'zipcode': '90566-7771', 'geo': {'lat': '-43.9509', 'lng': '-34.4618'}} | 010-692-6593 x09125 | anastasia.net | {'name': 'Deckow-Crist', 'catchPhrase': 'Proactive didactic contingency', 'bs': 'synergize scalable supply-chains'} |
| 2 | 3 | Clementine Bauch | Samantha | Nathan@yesenia.net | {'street': 'Douglas Extension', 'suite': 'Suite 847', 'city': 'McKenziehaven', 'zipcode': '59590-4157', 'geo': {'lat': '-68.6102', 'lng': '-47.0653'}} | 1-463-123-4447 | ramiro.info | {'name': 'Romaguera-Jacobson', 'catchPhrase': 'Face to face bifurcated interface', 'bs': 'e-enable strategic applications'} |
| 3 | 4 | Patricia Lebsack | Karianne | Julianne.OConner@kory.org | {'street': 'Hoeger Mall', 'suite': 'Apt. 692', 'city': 'South Elvis', 'zipcode': '53919-4257', 'geo': {'lat': '29.4572', 'lng': '-164.2990'}} | 493-170-9623 x156 | kale.biz | {'name': 'Robel-Corkery', 'catchPhrase': 'Multi-tiered zero tolerance productivity', 'bs': 'transition cutting-edge web services'} |
| 4 | 5 | Chelsey Dietrich | Kamren | Lucio\_Hettinger@annie.ca | {'street': 'Skiles Walks', 'suite': 'Suite 351', 'city': 'Roscoeview', 'zipcode': '33263', 'geo': {'lat': '-31.8129', 'lng': '62.5342'}} | (254)954-1289 | demarco.info | {'name': 'Keebler LLC', 'catchPhrase': 'User-centric fault-tolerant solution', 'bs': 'revolutionize end-to-end systems'} |
| 5 | 6 | Mrs. Dennis Schulist | Leopoldo\_Corkery | Karley\_Dach@jasper.info | {'street': 'Norberto Crossing', 'suite': 'Apt. 950', 'city': 'South Christy', 'zipcode': '23505-1337', 'geo': {'lat': '-71.4197', 'lng': '71.7478'}} | 1-477-935-8478 x6430 | ola.org | {'name': 'Considine-Lockman', 'catchPhrase': 'Synchronised bottom-line interface', 'bs': 'e-enable innovative applications'} |
| 6 | 7 | Kurtis Weissnat | Elwyn.Skiles | Telly.Hoeger@billy.biz | {'street': 'Rex Trail', 'suite': 'Suite 280', 'city': 'Howemouth', 'zipcode': '58804-1099', 'geo': {'lat': '24.8918', 'lng': '21.8984'}} | 210.067.6132 | elvis.io | {'name': 'Johns Group', 'catchPhrase': 'Configurable multimedia task-force', 'bs': 'generate enterprise e-tailers'} |
| 7 | 8 | Nicholas Runolfsdottir V | Maxime\_Nienow | Sherwood@rosamond.me | {'street': 'Ellsworth Summit', 'suite': 'Suite 729', 'city': 'Aliyaview', 'zipcode': '45169', 'geo': {'lat': '-14.3990', 'lng': '-120.7677'}} | 586.493.6943 x140 | jacynthe.com | {'name': 'Abernathy Group', 'catchPhrase': 'Implemented secondary concept', 'bs': 'e-enable extensible e-tailers'} |
| 8 | 9 | Glenna Reichert | Delphine | Chaim\_McDermott@dana.io | {'street': 'Dayna Park', 'suite': 'Suite 449', 'city': 'Bartholomebury', 'zipcode': '76495-3109', 'geo': {'lat': '24.6463', 'lng': '-168.8889'}} | (775)976-6794 x41206 | conrad.com | {'name': 'Yost and Sons', 'catchPhrase': 'Switchable contextually-based project', 'bs': 'aggregate real-time technologies'} |
| 9 | 10 | Clementina DuBuque | Moriah.Stanton | Rey.Padberg@karina.biz | {'street': 'Kattie Turnpike', 'suite': 'Suite 198', 'city': 'Lebsackbury', 'zipcode': '31428-2261', 'geo': {'lat': '-38.2386', 'lng': '57.2232'}} | 024-648-3804 | ambrose.net | {'name': 'Hoeger LLC', 'catchPhrase': 'Centralized empowering task-force', 'bs': 'target end-to-end models'} |

Este DataFrame possui 8 colunas: id, name (nome), username (nome de usuário), email , address (endereço), phone(telefone), website e company (empresa). Note que as colunas address e company contém informações aninhadas, ou seja, os dados dessas colunas estão organizados em subcampos. Por exemplo, na coluna address, você encontrará subcampos como street (rua), suite (complemento), city (cidade) e zipcode (CEP). Já na coluna company, os subcampos incluem name (nome da empresa), catchPhrase (slogan) e bs (área de atuação). Isso significa que, para acessar essas informações, é necessário realizar o processo de normalização.

**O desafio agora é normalizar esse DataFrame**, expandindo as colunas address e company em suas respectivas subcolunas para facilitar a visualização e a análise dos dados.

Se você tiver dúvidas sobre como resolver, consulte a opinião da pessoa instrutora!