

Projet Spark - Covid VS Société 2020

Problematique

Quelle est l'impacte du covid sur les société en France sur l'année 2020 ?

In [1]:

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql import *
import pyspark.sql.functions as f
from pyspark.sql.types import IntegerType
from pyspark.sql.functions import unix_timestamp, from_unixtime
from pyspark.sql.types import DoubleType, IntegerType
```

Création de la session spark

In [2]:

```
spark = SparkSession.builder.appName("covid_societe").getOrCreate()
```

Lecture du fichier des données covid-19

In [3]:

```
path_file_data = "../data/donnees_covid_societes/"

df_covid_par_dep = spark.read.csv(
    path_file_data + "covid/covid_quotidien_par_dep_du_patient/donnees-hospitalieres-nouv
eaux-covid19.csv",
    header=True,
    sep=";"
)

df_metadonne_covid_par_dep_et_sexe_patient = spark.read.csv(
    path_file_data + "covid/covid_quotidien_par_dep_du_patient/metadonnees-hospit-incid.c
sv",
    header=True,
    sep=";"
)

# Affichage des metadonnés
df_metadonne_covid_par_dep_et_sexe_patient.select(
    "Colonne",
    "Type ",
    "Description_FR"
).filter(
    df_metadonne_covid_par_dep_et_sexe_patient.Colonne.isNotNull()
).show(20, False)

# Affichage des données
df_covid_par_dep = df_covid_par_dep.filter(df_covid_par_dep.jour >= '01/01/2021')
df_covid_par_dep.show()

# La plus grande date
print("La plus grande date :")
df_covid_par_dep.orderBy('jour', ascending=False).show(1)
```

```
+-----+-----+-----+
|Colonne  |Type      |Description_FR      |
+-----+-----+-----+
|dep      |integer   |Département         |
|jour     |string($date)|Date de notification|
... ..
```

incid_hosp	string	Nombre quotidien de personnes nouvellement hospitalisees
incid_rea	integer	Nombre quotidien de nouvelles admissions en réanimation
incid_dc	integer	Nombre quotidien de personnes nouvellement décédées
incid_rad	integer	Nombre quotidien de nouveaux retours à domicile

dep	jour	incid_hosp	incid_rea	incid_dc	incid_rad
1	19/03/2020	1	0	0	0
2	19/03/2020	38	8	10	15
3	19/03/2020	2	0	0	6
4	19/03/2020	1	0	0	1
5	19/03/2020	4	0	0	1
6	19/03/2020	12	4	0	4
7	19/03/2020	7	1	0	0
8	19/03/2020	1	1	0	0
9	19/03/2020	0	0	0	0
10	19/03/2020	6	1	0	0
11	19/03/2020	19	5	3	4
12	19/03/2020	4	0	1	1
13	19/03/2020	89	16	0	43
14	19/03/2020	3	0	0	1
15	19/03/2020	1	0	0	0
16	19/03/2020	1	0	0	2
17	19/03/2020	2	0	0	0
18	19/03/2020	0	0	0	0
19	19/03/2020	1	0	0	0
21	19/03/2020	26	11	2	24

only showing top 20 rows

La plus grande date :

dep	jour	incid_hosp	incid_rea	incid_dc	incid_rad
1	31/12/2020	12	1	1	4

only showing top 1 row

Quelle departement a eu le plus d'hospitalisation, de réanimation et de décès ? (ecrire dans un csv dans /resultat)

In [4]:

```
df_covid_sum_par_dep = df_covid_par_dep.groupBy(
    "dep"
).agg(
    # On fait la somme de chaque colonne puis on cast le resultat en INTEGER et on le renomme
    f.sum("incid_hosp").cast(IntegerType()).alias('incid_hosp'),
    f.sum("incid_rea").cast(IntegerType()).alias('incid_rea'),
    f.sum("incid_rad").cast(IntegerType()).alias('incid_rad'),
    f.sum("incid_dc").cast(IntegerType()).alias('incid_dc')
).orderBy("dep",ascending=True)

# df_covid_sum_par_dep.show()

# On recupere les valeurs max
max_hospi = df_covid_sum_par_dep.groupBy().max("incid_hosp").collect()[0]["max(incid_hosp)"]
max_rea = df_covid_sum_par_dep.groupBy().max("incid_rea").collect()[0]["max(incid_rea)"]
max_dc = df_covid_sum_par_dep.groupBy().max("incid_dc").collect()[0]["max(incid_dc)"]
```

Département avec le plus d'hospitalisation

In [5]:

```
# Le département avec le plus d'hospitalisation
```

```
# Le département avec le plus d'hospitalisation
df_covid_max_hospi = df_covid_sum_par_dep.filter(
    f.col("incid_hosp") == max_hospi
).select(
    "dep",
    "incid_hosp"
)
df_covid_max_hospi.show()
```

```
+---+-----+
|dep|incid_hosp|
+---+-----+
| 75|      16559|
+---+-----+
```

Département avec le plus de réanimation

In [6]:

```
# Le département avec le plus de réanimation
df_covid_max_rea = df_covid_sum_par_dep.filter(
    f.col("incid_rea") == max_rea
).select(
    "dep",
    "incid_rea"
)
df_covid_max_rea.show()
```

```
+---+-----+
|dep|incid_rea|
+---+-----+
| 75|      3822|
+---+-----+
```

Département avec le plus de décès

In [7]:

```
# Le département avec le plus de décès
df_covid_max_dc = df_covid_sum_par_dep.filter(
    f.col("incid_dc") == max_dc
).select(
    "dep",
    "incid_dc"
)
df_covid_max_dc.show()
```

```
+---+-----+
|dep|incid_dc|
+---+-----+
| 75|      2850|
+---+-----+
```

Sauvergarder dans un fichier csv

In [8]:

```
path_file_save = "../resultat/"

# On est obliger de passer par pandas, sinon erreur
df_covid_max_hospi.toPandas().to_csv(path_file_save + 'departement_avec_le_plus_hospitali
sation.csv')
df_covid_max_rea.toPandas().to_csv(path_file_save + 'departement_avec_le_plus_de_reanimat
ion.csv')
df_covid_max_dc.toPandas().to_csv(path_file_save + 'departement_avec_le_plus_de_deces.csv
')
```

```
# df_covid_max_hospi.write.csv(path_file_save +'departement_avec_le_plus_hospitalisation.csv')
```

Réponse

Paris est la ville avec le plus d'incident:

- Hospitalisation: **16 559**
- Réanimation: **3 822**
- Décès: **2 850**

Quelle jour a été le plus critique en hospitalisation, reanimation et deces (par departement) (ecrire dans un csv dans /resultat)

DF1: dep | jour | max(hospi)

DF2: dep | jour | max(rea)

DF3: dep | jour | max(deces)

In [9]:

```
from pyspark.sql import Window

df_covid_max_par_dep_et_jour = df_covid_par_dep.withColumn("incid_hosp", df_covid_par_dep["incid_hosp"].cast(IntegerType()))
df_covid_max_par_dep_et_jour = df_covid_max_par_dep_et_jour.withColumn("incid_rea", df_covid_par_dep["incid_rea"].cast(IntegerType()))
df_covid_max_par_dep_et_jour = df_covid_max_par_dep_et_jour.withColumn("incid_dc", df_covid_par_dep["incid_dc"].cast(IntegerType()))
df_covid_max_par_dep_et_jour = df_covid_max_par_dep_et_jour.withColumn("incid_rad", df_covid_par_dep["incid_rad"].cast(IntegerType()))
```

In [10]:

```
w = Window.partitionBy('dep')
```

Le jour le plus critique en hospitalisation par département

In [11]:

```
df_max_hospi_jour_dep = df_covid_max_par_dep_et_jour.withColumn(
    'max',
    f.max('incid_hosp').over(w)
).where(
    f.col('incid_hosp') == f.col('max')
).select(
    "dep",
    "jour",
    "incid_hosp"
).orderBy(
    "dep",
    ascending=True
).drop("max")

df_max_hospi_jour_dep.show(120)
```

```
+---+-----+-----+
|dep|      jour|incid_hosp|
+---+-----+-----+
| 1|03/11/2020|      66|
|10|09/04/2020|     157|
|11|02/04/2020|      31|
|12|17/11/2020|      23|
|13|22/10/2020|     198|
|14|27/10/2020|      38|
```

15	20/11/2020	28
16	09/12/2020	15
17	19/01/2021	31
18	07/01/2021	33
19	30/10/2020	15
2	20/04/2020	106
21	25/03/2020	76
22	04/11/2020	16
22	02/04/2020	16
23	20/01/2021	19
24	23/03/2020	23
25	29/03/2020	68
26	04/11/2020	53
27	29/10/2020	28
28	02/04/2020	47
29	02/11/2020	17
29	11/11/2020	17
2A	19/03/2020	59
2B	20/03/2020	28
3	01/12/2020	38
30	04/11/2020	101
31	06/11/2020	59
32	05/05/2020	10
33	30/03/2020	89
34	29/10/2020	82
35	10/11/2020	46
36	21/04/2020	22
37	01/04/2020	31
37	04/11/2020	31
38	10/11/2020	166
39	03/11/2020	36
4	10/11/2020	30
40	13/11/2020	28
41	30/03/2020	23
41	30/04/2020	23
42	04/11/2020	104
43	05/11/2020	42
44	01/04/2020	45
45	12/11/2020	43
46	18/04/2020	26
47	09/12/2020	14
48	31/10/2020	15
49	31/03/2020	42
5	31/10/2020	53
50	19/01/2021	23
51	31/03/2020	100
52	03/04/2020	33
52	04/04/2020	33
53	18/11/2020	23
54	02/04/2020	101
55	30/03/2020	40
56	08/04/2020	35
57	24/03/2020	173
58	21/11/2020	28
59	06/11/2020	204
6	20/01/2021	68
60	02/04/2020	102
61	05/01/2021	31
62	07/11/2020	103
63	27/10/2020	66
64	04/11/2020	51
65	12/11/2020	25
66	22/03/2020	41
67	03/04/2020	165
68	25/03/2020	201
69	19/03/2020	244
7	28/04/2020	51
70	31/03/2020	27
71	03/11/2020	75
72	10/12/2020	36
73	04/11/2020	72
74	04/11/2020	95

17 03/03/2020	4
19 28/03/2020	3
19 01/04/2020	3
19 01/11/2020	3
2 19/03/2020	8
21 05/11/2020	17
22 02/04/2020	9
23 03/11/2020	6
24 02/11/2020	5
25 08/11/2020	18
26 30/03/2020	11
27 30/09/2020	6
27 23/04/2020	6
27 18/11/2020	6
28 23/03/2020	8
28 16/11/2020	8
29 28/03/2020	8
2A 19/03/2020	11
2B 12/01/2021	3
2B 12/05/2020	3
3 03/11/2020	8
30 04/11/2020	24
31 26/03/2020	19
32 20/01/2021	2
32 19/03/2020	2
32 20/04/2020	2
32 16/11/2020	2
33 24/03/2020	21
34 30/03/2020	14
34 21/10/2020	14
34 28/10/2020	14
35 19/03/2020	7
35 12/11/2020	7
35 31/03/2020	7
36 20/01/2021	3
36 23/03/2020	3
36 01/10/2020	3
36 27/03/2020	3
36 03/04/2020	3
36 28/03/2020	3
37 12/11/2020	12
38 13/11/2020	22
39 26/03/2020	5
39 21/12/2020	5
4 30/10/2020	5
40 15/01/2021	4
40 27/10/2020	4
40 05/01/2021	4
41 17/04/2020	6
42 28/03/2020	16
43 08/11/2020	6
44 31/03/2020	9
45 05/11/2020	14
45 03/04/2020	14
46 02/04/2020	3
46 19/11/2020	3
47 06/11/2020	6
48 19/01/2021	3
49 02/11/2020	12
5 31/10/2020	7
50 28/03/2020	4
50 08/01/2021	4
50 08/11/2020	4
51 30/03/2020	14
52 28/05/2020	9
53 16/01/2021	4
54 28/03/2020	26
55 03/04/2020	7
56 10/11/2020	8
57 28/03/2020	27
57 24/03/2020	27
58 05/04/2020	3

58 08/01/2020	5
59 31/03/2020	39
6 02/04/2020	17
60 02/04/2020	15
61 11/01/2021	4
62 30/03/2020	19
63 05/04/2020	11
63 30/03/2020	11
64 31/03/2020	8
65 06/01/2021	5
65 04/01/2021	5
66 19/03/2020	9
67 27/03/2020	36
67 01/04/2020	36
68 31/03/2020	37
69 28/03/2020	65
7 09/11/2020	5
7 31/03/2020	5
70 25/03/2020	8
70 31/03/2020	8
71 30/03/2020	11
71 13/11/2020	11
72 28/03/2020	6
72 30/03/2020	6
72 27/03/2020	6
73 10/11/2020	10
74 01/12/2020	13
75 02/04/2020	96
76 31/03/2020	20
77 31/03/2020	27
77 02/04/2020	27
77 26/03/2020	27
78 24/03/2020	23
79 18/11/2020	3
79 25/11/2020	3
79 29/03/2020	3

```
+---+-----+-----+
only showing top 120 rows
```

Le jour le plus critique en décès par département

In [13]:

```
df_max_deces_jour_dep = df_covid_max_par_dep_et_jour.withColumn(
    'max',
    f.max('incid_dc').over(w)
).where(
    f.col('incid_dc') == f.col('max')
).select(
    "dep",
    "jour",
    "incid_dc"
).orderBy(
    "dep",
    ascending=True
).drop("max")

df_max_deces_jour_dep.show(120)
```

```
+---+-----+-----+
|dep|      jour|incid_dc|
+---+-----+-----+
| 1|05/11/2020|      11|
| 1|24/11/2020|      11|
| 1|05/12/2020|      11|
| 1|14/01/2021|      11|
|10|09/04/2020|      11|
|11|28/10/2020|       6|
|12|29/10/2020|       5|
|12|17/11/2020|       5|
|12|19/11/2020|       5|
```


12 10/11/2020	5
13 02/11/2020	35
14 12/11/2020	8
15 03/12/2020	4
16 21/12/2020	5
17 13/11/2020	5
18 19/01/2021	10
19 04/04/2020	3
19 02/06/2020	3
19 02/11/2020	3
19 12/11/2020	3
2 03/04/2020	18
21 02/04/2020	18
22 24/12/2020	6
23 10/11/2020	4
24 07/12/2020	4
24 17/11/2020	4
25 31/03/2020	8
25 03/04/2020	8
25 14/04/2020	8
25 12/01/2021	8
26 20/11/2020	14
27 16/11/2020	13
28 22/12/2020	9
28 15/04/2020	9
29 01/01/2021	5
29 18/12/2020	5
2A 25/03/2020	7
2B 16/11/2020	3
2B 03/11/2020	3
3 24/11/2020	10
3 18/11/2020	10
30 04/05/2020	9
31 06/01/2021	12
32 23/04/2020	3
32 06/11/2020	3
32 20/01/2021	3
33 18/01/2021	11
34 09/11/2020	10
34 26/10/2020	10
34 23/11/2020	10
35 10/04/2020	9
36 13/04/2020	6
36 22/04/2020	6
36 19/11/2020	6
36 29/12/2020	6
37 01/04/2020	6
37 19/01/2021	6
38 13/11/2020	31
39 29/12/2020	13
4 03/12/2020	7
40 14/12/2020	8
41 07/12/2020	7
42 06/11/2020	19
42 09/11/2020	19
42 12/11/2020	19
43 09/12/2020	7
44 03/11/2020	11
45 05/11/2020	8
45 19/11/2020	8
46 02/05/2020	9
47 07/12/2020	5
48 03/12/2020	5
49 02/11/2020	8
49 17/11/2020	8
5 16/11/2020	8
50 10/11/2020	6
51 08/04/2020	17
52 03/04/2020	10
53 18/12/2020	6
53 29/12/2020	6
54 10/04/2020	24
55 28/12/2020	10

```

| 55|28/12/2020|10|
| 56|14/04/2020|5|
| 56|05/04/2020|5|
| 56|26/10/2020|5|
| 57|03/04/2020|42|
| 58|21/11/2020|11|
| 59|06/11/2020|36|
| 6|01/12/2020|17|
| 60|01/04/2020|20|
| 61|19/11/2020|9|
| 62|23/04/2020|22|
| 63|25/11/2020|11|
| 63|05/01/2021|11|
| 63|12/01/2021|11|
| 64|09/12/2020|12|
| 65|24/11/2020|7|
| 66|28/10/2020|7|
| 67|03/04/2020|30|
| 68|25/03/2020|52|
| 69|19/11/2020|37|
| 7|27/11/2020|10|
| 70|06/04/2020|7|
| 71|16/11/2020|17|
| 72|11/12/2020|9|
| 73|18/11/2020|16|
| 74|13/11/2020|15|
| 74|17/11/2020|15|
| 75|30/03/2020|69|
| 75|10/04/2020|69|
| 76|27/10/2020|16|
| 77|14/04/2020|29|
| 78|20/04/2020|23|
| 79|14/01/2021|8|
| 8|03/12/2020|8|
| 80|08/01/2021|16|
| 81|04/01/2021|6|
| 81|05/11/2020|6|
| 82|06/11/2020|4|
| 82|26/11/2020|4|
| 83|09/11/2020|15|
+---+-----+-----+

```

only showing top 120 rows

Sauvergarder dans un fichier csv

In [14]:

```

path_file_save = "../resultat/"

# On est obliger de passer par pandas, sinon erreur
df_max_hospi_jour_dep.toPandas().to_csv(path_file_save + 'jour_critique_hospitalisation.csv')
df_max_rea_jour_dep.toPandas().to_csv(path_file_save + 'jour_critique_reanimation.csv')
df_max_deces_jour_dep.toPandas().to_csv(path_file_save + 'jour_critique_deces.csv')

```

Lecture du fichier des société radiees

In [15]:

```

path_file_data = "../data/donnees_covid_societes/"

df_societe_radie = spark.read.csv(
    path_file_data + "societe/societes-radiees-2020.csv",
    header=True,
    sep=";"
)
# df_societe_radie.show()

```

```
# On recupere uniquement les colonnes qui nous interesse
df_societe_radie = df_societe_radie.select(
    "Dénomination",
    "Num dept",
    "Département",
    "Date radiation"
)

# On Supprime les lignes avec des valeur vide
df_societe_radie = df_societe_radie.filter(
    f.col("Num dept").isNotNull() &
    f.col("Date radiation").isNotNull()
).orderBy('Date radiation', ascending=True)

# Affichage des données
df_societe_radie.show()

# La plus grande date
print("La plus grande date :")
df_societe_radie.select("Date radiation").orderBy('Date radiation', ascending=False).show(1)
```

```
+-----+-----+-----+-----+
|      Dénomination|Num dept|      Département|Date radiation|
+-----+-----+-----+-----+
|      SEBAFLEX|      21|      Côte d'Or|      01/02/2020|
|JEROME RAVET CONSEIL|      92|      Hauts-de-Seine|      01/02/2020|
|      JLD TRAITEUR|      42|      Loire|      01/02/2020|
|      LA CHOUETTE VTC|      21|      Côte d'Or|      01/02/2020|
|      AND CO|      92|      Hauts-de-Seine|      01/02/2020|
|      L'ART A 4 MAINS|      21|      Côte d'Or|      01/02/2020|
|      COMP-I|      61|      Orne|      01/02/2020|
|SOCIETE SHIRLEY P...|      92|      Hauts-de-Seine|      01/02/2020|
|      SPORT KIFF|      21|      Côte d'Or|      01/02/2020|
|      ARMICOM|      44|Loire-Atlantique|      01/02/2020|
|      COPOPLAST|      21|      Côte d'Or|      01/02/2020|
|      HIGH TECH FINANCE|      42|      Loire|      01/02/2020|
|CHATEAU LANDON EN...|      77|      Seine-et-Marne|      01/03/2020|
|      E.R.C|      77|      Seine-et-Marne|      01/03/2020|
|      ROYAL REPTILE|      77|      Seine-et-Marne|      01/03/2020|
|      EASY JUST|      77|      Seine-et-Marne|      01/03/2020|
|      PRS|      77|      Seine-et-Marne|      01/03/2020|
|      BASABOZ|      77|      Seine-et-Marne|      01/03/2020|
|      LE PLOMBIER|      77|      Seine-et-Marne|      01/03/2020|
|      GSS SECURITE|      77|      Seine-et-Marne|      01/03/2020|
+-----+-----+-----+-----+
```

only showing top 20 rows

La plus grande date :

```
+-----+
|Date radiation|
+-----+
|      31/12/2020|
+-----+
```

only showing top 1 row

Quelle département à eu le plus de radiation d'entreprise en 2020 ?

DF_final: Num dept | nombre_dentreprise_raddie

In [16]:

```
# Nombre de societe radié par departement et par jour
df_nb_societe_radie = df_societe_radie.groupBy(
    "Num dept",
    "Date radiation",
    from_unixtime(unix_timestamp("Date radiation", 'dd/MM/yyyy')).alias('Date radiation un
ix')
).agg(
```

```
# On fait le count de chaque colonne puis on cast le resultat en INTEGER et on le renomme
```

```
f.count("Num dept").cast(IntegerType()).alias('nb_societe_radie')
).orderBy("Num dept", f.col("nb_societe_radie").desc())
```

```
df_nb_societe_radie.show()
```

```
+-----+-----+-----+-----+
|Num dept|Date radiation|Date radiation unix|nb_societe_radie|
+-----+-----+-----+-----+
|      1| 16/12/2020|2020-12-16 00:00:00|          208|
|      1| 24/12/2020|2020-12-24 00:00:00|           98|
|      1| 28/09/2020|2020-09-28 00:00:00|           85|
|      1| 23/09/2020|2020-09-23 00:00:00|           67|
|      1| 18/09/2020|2020-09-18 00:00:00|           61|
|      1| 09/10/2020|2020-10-09 00:00:00|           48|
|      1| 18/06/2020|2020-06-18 00:00:00|           43|
|      1| 10/12/2020|2020-12-10 00:00:00|           35|
|      1| 05/05/2020|2020-05-05 00:00:00|           32|
|      1| 16/01/2020|2020-01-16 00:00:00|           32|
|      1| 12/03/2020|2020-03-12 00:00:00|           31|
|      1| 15/12/2020|2020-12-15 00:00:00|           30|
|      1| 12/06/2020|2020-06-12 00:00:00|           29|
|      1| 04/12/2020|2020-12-04 00:00:00|           27|
|      1| 10/11/2020|2020-11-10 00:00:00|           25|
|      1| 29/10/2020|2020-10-29 00:00:00|           24|
|      1| 19/11/2020|2020-11-19 00:00:00|           22|
|      1| 22/09/2020|2020-09-22 00:00:00|           21|
|      1| 09/07/2020|2020-07-09 00:00:00|           20|
|      1| 21/02/2020|2020-02-21 00:00:00|           19|
+-----+-----+-----+-----+
```

only showing top 20 rows

In [17]:

```
df_dpt_radie = df_nb_societe_radie.groupBy(
    "Num dept"
).agg(
    # On fait la somme du nombre d'entreprises radiées par département
    f.sum("nb_societe_radie").cast(IntegerType()).alias("nombre_dentreprise_radie")
)

# On récupère la valeur maximale d'entreprises radiées
max_radie_par_dpt = df_dpt_radie.groupBy().max("nombre_dentreprise_radie").collect()[0][
    "max(nombre_dentreprise_radie)"]

df_dpt_radie_max = df_dpt_radie.filter(
    # On filtre par la valeur maximale d'entreprises radiées
    f.col("nombre_dentreprise_radie") == max_radie_par_dpt
).select(
    "Num dept",
    "nombre_dentreprise_radie"
)

print("Département avec le plus de radiation d'entreprise:")
df_dpt_radie_max.show()
```

Département avec le plus de radiation d'entreprise:

```
+-----+-----+
|Num dept|nombre_dentreprise_radie|
+-----+-----+
|      75|          16897|
+-----+-----+
```

Quelle est l'impacte du covid sur les société en France sur l'année 2020 ?

DF societe : Num dept | date radiation | nb_de_societe_radié

DF covid : dep | jour | hosp | rea | rad | dc

Df covid_societe : dep | date | nb_de_societe_radié | hosp | rea | rad | dc

In [18]:

```
# Nombre de societe radié par departement et par jour
df_nb_societe_radie = df_societe_radie.groupBy(
    "Num dept",
    "Date radiation",
    from_unixtime(unix_timestamp("Date radiation", 'dd/MM/yyyy')).alias('Date radiation unix')
).agg(
    # On fait le count de chaque colonne puis on cast le resultat en INTEGER et on le renomme
    f.count("Num dept").cast(IntegerType()).alias('nb_societe_radie')
).orderBy("Num dept", "Date radiation unix")

# Ajout de la date au format unix sur le dataframe de covid
df_covid_unix = df_covid_par_dep.select(
    "dep", "jour", "incid_hosp", "incid_rea", "incid_dc", "incid_rad",
    from_unixtime(unix_timestamp("jour", 'dd/MM/yyyy')).alias('jour unix')
).orderBy("dep", "jour unix")

df_covid_unix.show()
```

dep	jour	incid_hosp	incid_rea	incid_dc	incid_rad	jour unix
1	19/03/2020	1	0	0	0	2020-03-19 00:00:00
1	20/03/2020	0	0	0	1	2020-03-20 00:00:00
1	21/03/2020	3	0	0	0	2020-03-21 00:00:00
1	22/03/2020	3	1	0	1	2020-03-22 00:00:00
1	23/03/2020	14	1	0	5	2020-03-23 00:00:00
1	24/03/2020	11	1	0	4	2020-03-24 00:00:00
1	25/03/2020	13	2	0	5	2020-03-25 00:00:00
1	26/03/2020	14	3	2	2	2020-03-26 00:00:00
1	27/03/2020	14	2	0	0	2020-03-27 00:00:00
1	28/03/2020	7	3	1	3	2020-03-28 00:00:00
1	29/03/2020	11	3	3	3	2020-03-29 00:00:00
1	30/03/2020	20	7	4	1	2020-03-30 00:00:00
1	31/03/2020	20	0	1	9	2020-03-31 00:00:00
1	01/04/2020	38	14	2	10	2020-04-01 00:00:00
1	02/04/2020	34	7	2	19	2020-04-02 00:00:00
1	03/04/2020	15	4	3	13	2020-04-03 00:00:00
1	04/04/2020	25	8	5	9	2020-04-04 00:00:00
1	05/04/2020	12	4	2	3	2020-04-05 00:00:00
1	06/04/2020	11	3	1	6	2020-04-06 00:00:00
1	07/04/2020	15	3	2	5	2020-04-07 00:00:00

only showing top 20 rows

In [19]:

```
# Jointure entre le df des societés et du covid
df_societe_radie_join_covid = df_covid_unix.join(
    df_nb_societe_radie,
    (df_covid_unix.dep == f.col("Num dept")) & (f.col("jour unix") == f.col("Date radiation unix"))
)

# On recupere les colonnes qui nous intéresse
df_societe_radie_join_covid = df_societe_radie_join_covid.select(
    "dep",
    "jour unix",
    "incid_hosp",
    "incid_rea",
    "incid_dc",
    "incid_rad",
    "nb_societe_radie"
).orderBy("dep", "jour unix")
```

```
df_societe_radie_join_covid.show()
```

dep	jour unix	incid_hosp	incid_rea	incid_dc	incid_rad	nb_societe_radie
1	2020-03-23 00:00:00	14	1	0	5	2
1	2020-03-24 00:00:00	11	1	0	4	2
1	2020-03-25 00:00:00	13	2	0	5	4
1	2020-03-26 00:00:00	14	3	2	2	1
1	2020-03-30 00:00:00	20	7	4	1	4
1	2020-03-31 00:00:00	20	0	1	9	1
1	2020-04-03 00:00:00	15	4	3	13	1
1	2020-04-06 00:00:00	11	3	1	6	1
1	2020-04-07 00:00:00	15	3	2	5	2
1	2020-04-14 00:00:00	14	1	2	5	1
1	2020-04-15 00:00:00	35	0	6	28	2
1	2020-04-16 00:00:00	16	1	0	3	2
1	2020-04-17 00:00:00	9	2	3	9	1
1	2020-04-20 00:00:00	4	2	2	3	2
1	2020-04-21 00:00:00	12	3	2	11	4
1	2020-04-23 00:00:00	10	0	4	9	2
1	2020-04-24 00:00:00	7	1	1	5	1
1	2020-04-27 00:00:00	4	0	0	7	5
1	2020-04-28 00:00:00	4	0	3	4	4
1	2020-04-29 00:00:00	14	0	1	11	1

only showing top 20 rows

In [20]:

```
# Nombre de incident covid et de société radié (par département et par mois)
df_groupby_par_dep_mois_societe_covid = df_societe_radie_join_covid.groupBy(
    f.col("dep").cast(IntegerType()).alias('dep'),
    f.month(f.col("jour unix")).alias("jour unix month")
).agg(
    f.sum("incid_hosp").cast(IntegerType()).alias('incid_hosp'),
    f.sum("incid_rea").cast(IntegerType()).alias('incid_rea'),
    f.sum("incid_dc").cast(IntegerType()).alias('incid_dc'),
    f.sum("incid_rad").cast(IntegerType()).alias('incid_rad'),
    f.sum("nb_societe_radie").cast(IntegerType()).alias('nb_societe_radie'),
).orderBy("dep", "jour unix month")
```

```
df_groupby_par_dep_mois_societe_covid.show(150)
```

dep	jour unix month	incid_hosp	incid_rea	incid_dc	incid_rad	nb_societe_radie
1	3	92	14	7	26	14
1	4	178	22	31	127	31
1	5	45	5	9	85	76
1	6	16	1	6	31	161
1	7	5	2	2	13	54
1	8	9	0	0	10	27
1	9	48	8	4	27	290
1	10	301	29	20	125	127
1	11	631	55	124	395	98
1	12	207	27	47	208	472
2	3	127	14	12	74	5
2	4	167	12	32	93	10
2	5	97	8	20	118	14
2	6	16	4	3	34	182
2	7	23	3	3	58	82
2	8	11	3	1	11	8
2	9	20	3	1	17	34
2	10	130	18	9	58	46
2	11	335	40	49	258	52
2	12	331	40	79	206	69
3	3	20	2	3	8	6
3	4	15	4	1	10	6
3	5	9	0	5	15	32

3	6	17	2	1	10	23
3	7	2	0	0	5	33
3	8	8	0	0	4	6
3	9	22	2	3	16	30
3	10	148	16	16	63	42
3	11	295	31	76	194	28
3	12	178	8	47	125	46
4	3	5	0	1	3	1
4	4	24	1	1	20	9
4	5	0	0	2	3	31
4	6	1	0	1	6	23
4	7	1	0	0	0	32
4	8	3	0	0	3	16
4	9	13	0	1	13	40
4	10	38	1	4	16	17
4	11	197	7	16	144	30
4	12	85	1	15	76	45
5	3	9	1	0	3	2
5	4	6	0	0	9	7
5	5	2	0	2	2	15
5	6	2	1	1	8	23
5	7	0	0	0	2	25
5	8	0	0	0	1	6
5	9	16	2	1	8	47
5	10	93	8	7	29	40
5	11	103	14	12	65	13
5	12	75	14	26	75	37
6	3	171	31	14	64	347
6	4	432	117	77	249	78
6	5	105	13	32	147	111
6	6	31	1	9	75	213
6	7	41	6	2	40	285
6	8	156	14	3	98	118
6	9	278	43	24	212	236
6	10	399	77	26	246	344
6	11	724	119	124	458	310
6	12	719	103	161	492	722
7	4	72	6	9	57	10
7	5	80	1	18	110	21
7	6	3	1	1	10	26
7	7	4	0	2	4	36
7	8	19	0	2	5	16
7	9	13	3	3	20	206
7	10	91	8	10	36	37
7	11	174	11	42	152	31
7	12	163	17	35	176	54
8	3	0	0	0	0	1
8	4	3	0	0	6	2
8	5	24	1	1	17	30
8	6	0	0	1	14	7
8	7	1	0	0	0	67
8	8	2	0	0	2	7
8	9	9	1	0	5	41
8	10	21	6	1	12	22
8	11	150	8	29	52	17
8	12	94	13	34	85	45
9	3	3	0	0	2	1
9	4	8	3	0	7	5
9	5	2	0	1	2	10
9	6	0	0	0	1	14
9	7	0	0	0	0	10
9	8	0	0	0	0	5
9	9	2	1	0	3	14
9	10	15	2	2	6	17
9	11	16	3	3	10	12
9	12	10	2	4	6	29
10	3	50	6	7	4	13
10	4	70	4	14	49	37
10	5	44	0	10	63	45
10	6	11	1	0	16	19
10	7	5	2	0	5	46
10	8	2	0	1	7	12

10	9	23	5	1	22	34
10	10	54	6	3	21	41
10	11	88	12	20	57	45
10	12	62	11	12	84	18
11	3	53	8	5	4	8
11	4	28	4	5	29	9
11	5	10	1	1	26	46
11	6	7	0	2	9	47
11	7	1	0	0	2	23
11	8	2	1	0	2	51
11	9	13	3	2	13	64
11	10	77	9	15	40	43
11	11	112	8	30	82	60
11	12	51	2	11	30	32
12	3	27	6	4	6	10
12	4	8	0	2	30	7
12	5	1	0	1	5	18
12	6	0	0	0	0	29
12	7	1	0	0	0	46
12	8	3	0	0	1	11
12	9	16	3	0	9	19
12	10	51	7	7	24	23
12	11	126	13	26	90	41
12	12	64	4	4	59	27
13	3	853	138	33	346	43
13	4	1962	313	290	1433	137
13	5	333	55	92	555	187
13	6	102	13	30	269	296
13	7	132	14	9	182	376
13	8	414	44	16	431	334
13	9	1393	243	77	1055	551
13	10	2283	342	221	1390	347
13	11	2175	379	421	1939	395
13	12	1421	268	261	1299	377
14	3	65	20	2	14	12
14	4	80	17	12	34	15
14	5	29	1	13	68	29
14	6	5	0	0	24	80
14	7	6	1	2	9	54
14	8	2	1	0	16	111
14	9	39	9	2	19	34
14	10	218	39	20	97	91
14	11	352	53	68	235	96
14	12	176	35	55	195	113
15	3	0	0	0	0	3
15	4	6	1	1	13	7
15	5	1	1	0	3	5
15	6	0	0	0	1	16
15	7	0	0	0	2	10
15	8	0	0	0	0	7
15	9	7	0	2	4	14
15	10	31	7	1	13	17
15	11	36	7	2	22	28
15	12	40	4	11	41	17
16	3	12	4	0	2	2

only showing top 150 rows

In [21]:

```
# Nombre d'incident covid et de société radié par mois
df_groupby_par_mois_societe_covid = df_societe_radie_join_covid.groupby(
    f.month(f.col("jour unix")).alias("month")
).agg(
    f.sum("incid_hosp").cast(IntegerType()).alias('incid_hosp'),
    f.sum("incid_rea").cast(IntegerType()).alias('incid_rea'),
    f.sum("incid_dc").cast(IntegerType()).alias('incid_dc'),
    f.sum("incid_rad").cast(IntegerType()).alias('incid_rad'),
    f.sum("nb_societe_radie").cast(IntegerType()).alias('nb_societe_radie'),
).orderBy("month")
```



```
df_groupby_par_mois_societe_covid.show(150)
```

```
+-----+-----+-----+-----+-----+-----+
|month|incid_hosp|incid_rea|incid_dc|incid_rad|nb_societe_radie|
+-----+-----+-----+-----+-----+-----+
| 3|16735|3435|1540|4602|2595|
| 4|32671|4925|6433|23017|5343|
| 5|7177|911|1933|12424|6558|
| 6|2453|312|671|6069|12025|
| 7|1896|281|315|4126|11340|
| 8|3209|515|283|3339|7693|
| 9|10968|1987|909|7871|14976|
| 10|29940|4911|2805|15580|11476|
| 11|41105|6440|7549|31016|11224|
| 12|26669|3905|5955|23985|17398|
+-----+-----+-----+-----+-----+-----+
```

In [22]:

```
# Nombre d'incident covid et de société radié par département
df_groupby_par_dep_societe_covid = df_societe_radie_join_covid.groupBy(
    f.col("dep").cast(IntegerType()).alias('dep'),
).agg(
    f.sum("incid_hosp").cast(IntegerType()).alias('incid_hosp'),
    f.sum("incid_rea").cast(IntegerType()).alias('incid_rea'),
    f.sum("incid_dc").cast(IntegerType()).alias('incid_dc'),
    f.sum("incid_rad").cast(IntegerType()).alias('incid_rad'),
    f.sum("nb_societe_radie").cast(IntegerType()).alias('nb_societe_radie'),
).orderBy("dep")
```

```
df_groupby_par_dep_societe_covid.show(150)
```

```
+---+-----+-----+-----+-----+-----+
|dep|incid_hosp|incid_rea|incid_dc|incid_rad|nb_societe_radie|
+---+-----+-----+-----+-----+-----+
| 1|1532|163|250|1047|1350|
| 2|1257|145|209|927|502|
| 3|714|65|152|450|252|
| 4|367|10|41|284|244|
| 5|306|40|49|202|215|
| 6|3056|524|472|2081|2764|
| 7|619|47|122|570|437|
| 8|304|29|66|193|239|
| 9|56|11|10|37|117|
| 10|409|47|68|328|310|
| 11|354|36|71|237|383|
| 12|297|33|44|224|231|
| 13|11068|1809|1450|8899|3043|
| 14|972|176|174|711|635|
| 15|121|20|17|99|124|
| 16|163|26|28|114|295|
| 17|456|73|85|326|730|
| 18|380|22|104|288|190|
| 19|169|26|22|152|191|
| 21|1436|258|259|1050|653|
| 22|395|49|56|294|396|
| 23|39|11|9|49|65|
| 24|301|43|37|226|371|
| 25|809|173|157|650|430|
| 26|1292|195|241|933|617|
| 27|807|86|173|599|538|
| 28|706|99|114|622|369|
| 29|450|95|65|309|543|
| 30|1352|334|229|1049|1684|
| 31|2452|591|283|1970|2408|
| 32|112|11|27|91|188|
| 33|2977|551|376|2182|2657|
| 34|2478|513|376|1905|1861|
| 35|1351|195|209|971|1210|
| 36|111|13|33|112|122|
| 37|785|143|146|537|833|
+---+-----+-----+-----+-----+-----+
```

38	4152	692	745	2953	1834
39	455	34	89	253	191
40	434	41	72	262	378
41	523	61	78	345	291
42	3165	375	609	2417	1017
43	423	32	78	290	218
44	1850	288	302	1491	1836
45	1203	332	136	863	831
46	78	7	8	77	131
47	247	38	43	202	463
48	27	1	9	15	117
49	1362	241	211	964	572
50	425	52	83	336	337
51	1520	189	241	1033	649
52	181	22	25	92	89
53	437	41	66	300	297
54	2272	367	381	1492	701
55	273	20	47	143	133
56	751	99	126	559	736
57	1813	250	462	1657	750
58	132	13	28	135	118
59	9188	1646	1567	7032	2847
60	1733	226	375	1309	784
61	361	26	54	225	156
62	3658	579	632	2911	1171
63	1218	175	223	812	571
64	1250	135	212	872	747
65	256	28	41	190	264
66	432	91	73	363	692
67	2131	267	393	2089	1240
68	1494	207	223	1718	689
69	10034	1672	1509	7905	3066
70	185	31	28	179	134
71	2196	206	387	1620	399
72	1028	81	131	700	472
73	1498	190	294	1192	862
74	2665	368	447	2187	1383
75	11983	2671	2064	9135	12371
76	2749	419	503	1858	909
77	5572	1030	936	4019	2313
78	5491	606	922	3925	1884
79	219	21	36	182	263
80	912	152	204	690	388
81	583	138	92	385	473
82	217	52	44	188	301
83	2472	387	392	2006	1559
84	1510	133	293	1011	970
85	590	86	80	412	1348
86	354	50	82	290	308
87	531	21	81	317	307
88	992	101	204	655	305
89	820	93	135	528	444
90	316	51	71	200	172
91	4877	782	773	3912	3009
92	8878	1696	1263	6903	3874
93	7824	1254	1178	6174	5834
94	7997	1188	1389	6307	2194
95	4751	539	837	3555	1779
971	509	125	91	435	478
972	280	79	25	207	3064
973	317	25	9	298	181
974	556	135	35	512	1527
976	20	3	2	24	10
+---+-----+-----+-----+-----+-----+					

Sauvergarder dans un fichier csv

In [23]:

```
path file save = "../resultat/"
```

```
# On est obliger de passer par pandas, sinon erreur
```

```
df_groupby_par_dep_mois_societe_covid.toPandas().to_csv(path_file_save + 'Nombre dinciden  
t covid et de société radié par département et par mois.csv')  
df_groupby_par_mois_societe_covid.toPandas().to_csv(path_file_save + 'Nombre dincident co  
vid et de société radié par mois.csv')  
df_groupby_par_dep_societe_covid.toPandas().to_csv(path_file_save + 'Nombre dincident cov  
id et de société radié par département.csv')
```

Datavisualisation

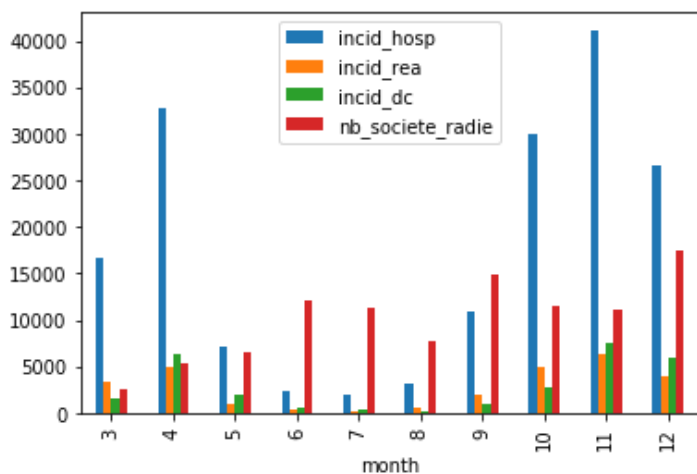
In [24]:

```
import matplotlib.pyplot as plt
```

Nombre d'incident covid et de société radié par mois

In [25]:

```
df_datavis = df_groupby_par_mois_societe_covid.toPandas()  
df_datavis.plot(kind='bar', x='month', y=["incid_hosp", "incid_rea", "incid_dc", "nb_soc  
iete_radie"])  
plt.show()
```



Conclusion

Pour conclure, nous pouvons constaté avec ce graphique que le covid a eu un impacte sur les dsociete en France en 2020.

Nous voyons ici que l'impacte est effectif 2 mois apres les piques

Par exemple en mars/ avril il y a une augmentation d'hospitalisation ce qui est répercuter 2 mois apres avec l'augmentation du nombre de societe radié.

OUI le covid a eu un impacte sur les societe en France