

Comparing Core Pyspark and Pandas Code Cheat Sheet



Importing Dataset	View number of columns and rows	Display column datatypes	Filter comparisons
<pre>#SPARK #PANDAS #FromPandas titanic_sp titanic_pd = spark.titanic - titanic = pandas.titanic ab le(" c_s p.s e pysparkDF2len(ti tan ic_sp tit - le - .c ol - d.shape ani c_t r ct(" *").t ai - oP - = spark.c re -umns))) n") andas() aFr - ame (pa ndasDF)</pre>	<pre>#SPARK print((ti - #PANDAS tan ic_sp .c titani c_p ou nt(), - d.shape re -umns)))</pre>	<pre>titani c_s p. titani c_p d.d d - - types types</pre>	<pre>df[df['sp ec i - es'].i sin ([' Chi - ' Chi - nst rap', nst rap', 'Gento o'])] 'Gento o'])]. .s - h - how(5) ead()</pre>
View data in DataFrame	Dropping Columns	flight data =	df[df['sp ec i - es'].r lik e es'].s tr.m (' - at - G.')].s how ch('G.')].he ad(-))
<pre>titani c_s displa y(t ita p.s - - how() nic_pd)</pre>	<pre>flight data = flight data = flight da ta.d flight da ta. ro - d - p(c olu mns tro p(* col o - umn - _drop, axis = 1) s_t o_drop)</pre>	<pre>flight da ta['dt - de par tur e'] = hCo lum n(' pd.to_ dat e dt_ - ti - me(fli ght dep art ure', da - f.to_t ime st ta['dt de a - par - mp(fli ght tur e_d ate da - - ta.d ep art time'], ure - format - ='% Y-%m- _da tetime, %d 'yyyy- MM-dd %H%M') HHmm'))</pre>	<pre>df[df['fl ipp df[df['fl ipp - - er'].b etw er'].b etw een - een - (22 5,2 29) (22 5,2 29)].s -].h - how(5) ead()</pre>
Display DataFrame Schema	Unique values of a column	Summary Stats	df[df['ma s s'].i - sNu ll()].s sNu ll()].h ead()
<pre>titani c_s p.p rin tSc - titani c_p d.i - hema() nfo() The column names, column data type, non-null values and Pandas memory use</pre>	<pre>titani c_s p.s ele - titani c_p d[' - ct('Su rvi ved ').d is - Sur viv - tin ct().s how() ed'].u - nique()</pre>	<pre>df.des - df.des cri be(). s - cribe() how()</pre>	<pre>df[(df ['m as s ']< - 3400) & (df['s - ex']== 'Ma - le')].h ead()) df[(df ['f lip - per '].b et wee - n(2 25, 22 9).s - how(5))]</pre>
Renaming a column in	Unique values of a column	Aggregation	flight da - flight data =
<pre>column re nam - titani c_p d.r - ed= tit ani c_s p.w - ena me(col - ith Col umn Ren - umn s= ame d("N - { 'Name': ame " ,"Pa sse - 'Passe nge - nge rNa me").co - rNa me' }).c - lumns olumns</pre>	<pre>titani c_s p.c - titani c_p d.c - olumns olumns</pre>	<pre>pyspar kDF.gr o df.gro - up - upB y("C By(" gen der - ") \ .agg(m ean ("ag omp - e"), - any "). mea n("s ala ry a - ") ,ma - gg({'S ale - x("s ala ry")) \ s: 'su - .show() m'}).s - how()</pre>	<pre>flight da ta['de par flight da ta.w tur - it - e_d ela y_shCo lum n(' - dep - art ure de la tatus'] = y st - np.whe re(atus',</pre>

```
fli - f.when (fl -  
      igh t_d ata.de  
ght da - par -
```

```
ta[ 'de par tur e_delay >  
tur - 90,  
e_d elay'] > 'Heavy ').o th
```

```
erw -  
ise ('M ode r  
90, 'Heavy', ate)
```

'Moder ate')