

Importing Dataset <pre>#SPARK #PANDAS #FromP- titanic_sp titanic_pd dBackT- = spark.t = titani- oSPARK able("tit- c_sp.sele- pysparkDF2en(titanic_sp.col- anic_train") andas() = spark.cre-umsn))) ateDataFr-</pre>	View number of columns and rows <pre>#SPARK print((ti- #PANDAS tanic_sp.count(), titanic_p- 2en(titanic_sp.col- d.shape umsn)))</pre>	Display column datatypes <pre>titanic_sp.d- titanic_pd.d- types types</pre>	Filter comparisons <pre>df[df['speci- df[df['speci- es'].isin(['Chi- es'].isin(['Chi- nstrap', nstrap', 'Gentoo'])].s- 'Gentoo'])].h- how(5) ead()</pre>
View data in DataFrame <pre>titanic_sp.s- display(tita- how() nic_pd)</pre>	Dropping Columns <pre>flight_data = flight_data = flight_data.dro- flight_data.d- p(columns_to- rop(*column- _drop, axis = 1) s_to_drop)</pre>	Convert tyoes <pre>flight_data = flight_data['dt- flight_data.wit- _departure'] = hColumn('dt- pd.to_dateti- departure', me(flight_da- f.to_timesta- ta['dt_depar- mp(flight_da- ture_date- ta.departure- time'], format- _datetime, ='%Y-%m-%d 'yyyy-MM-dd %H%M')) HHmm'))</pre>	<pre>df[df['speci- df[df['speci- es'].rlike('- es'].str.mat- G.').show(5) ch('G.').head(-)</pre>
Display DataFrame schema <pre>titanic_sp.printSc- titanic_pd.i- hema() nfo()</pre> <p>The column names, column data type, non-null values and Pandas memory use</p>	Unique values of a column <pre>titanic_sp.sele- titanic_pd['- ct('Survived').dis- Surviv- tinct().show() ed'].u- nique()</pre>	Summary Stats <pre>df.des- df.describe().s- cribe() how()</pre>	<pre>df[df['mass'].i- df[df['mass'].i- sNull()].s- snull()).head() how(5)</pre>
Renaming a column in a DataFrame <pre>column_renam- titanic_pd.r- ed=titanic_sp.w- ename(col- ithColumnRen- umns= amed("N- {'Name': ame","Passe- 'Passenge- ngerName").co- rName'}).c- lumns olumns</pre>	View column names <pre>titanic_sp.c- titanic_pd.c- olumns olumns</pre>	Aggregation <pre>df.gro- pysparkDF.group- upBy("C- By("gender") \ omp- .agg(mean("age"),- any").a- mean("salary"),ma- gg({'Sale- x("salary")}) \ s':'su- .show() m'}).s- how()</pre>	<pre>df[(df['mass']<- df[(df['mass']<- 3400) & (df['s- 3400) & (df['s- ex']=='Ma- ex']=='Ma- le')].head() le')].show(5)</pre>
			Conditional Transformations <pre>flight_da- flight_data = ta['departur- flight_data.wit- e_delay_s- hColumn('dep- tatus'] = arture_delay_st- np.where(fli- atus', f.when(fli- ght_da- ight_data.depar- ta['departur- ture_delay > 90, e_delay'] > 'Heavy').otherw- 90, 'Heavy', ise('Moderate') 'Moderate')</pre>



By **datamansam**

Published 3rd May, 2022.
Last updated 28th May, 2022.
Page 1 of 1.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>