

Importing Dataset

```
#SPARK #PANDAS #FromP-
titanic_sp titanic_pd dBackT-
= spark.t = titani- oSPARK
able("tit- c_sp.sele- pysparkDF2en(titanic_sp.col-
anic_trai- ct("**").toP- = spark.cre-umns)))
n") andas() ateDataFr-
ame(pa-
ndasDF)
```

View data in DataFrame

```
titanic_sp.s- display(tita-
how() nic_pd)
```

Display DataFrame schema

```
titanic_sp.printSc- titanic_pd.i-
hema() nfo()
```

The column names, column data type, non-null values and Pandas memory use

Renaming a column in a DataFrame

```
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
```

View number of columns and rows

```
#SPARK print((ti- #PANDAS
tanic_sp.count(), titanic_p-
2en(titanic_sp.col- d.shape
umns)))
```

Dropping Columns

```
flight_data = flight_data =
flight_data.dro- flight_data.d-
p(columns_to- rop(*column-
_drop, axis = 1) s_to_drop)
```

Unique values of a column

```
titanic_sp.sele- titanic_pd['-
ct('Survived').dis- Surviv-
tinct().show() ed'].u-
nique()
```

View column names

```
titanic_sp.c- titanic_pd.c-
olumns olumns
```

Display column datatypes

```
titanic_sp.d- titanic_pd.d-
types types
```

Convert tyoes

```
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
'yyyyMMdd HHmm'))
```

Summary Stats

```
df.des- df.describe().s-
cribe() how()
```

Aggregation

```
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()
```

Filter comparisons

```
df[df['speci- df[df['speci-
es'].isin(['Chi- es'].isin(['Chi-
nstrap', nstrap',
'Gentoo'])].s- 'Gentoo'])].h-
how(5) ead()
```

```
df[df['speci- df[df['speci-
es'].rlike('- es'].str.mat-
G.').show(5) ch('G.').head(-
)]
```

```
df[df['flipp- df[df['flipp-
er'].between- er'].between-
(225,229)].s- (225,229)].h-
how(5) ead()
```

```
df[df['mass'].i- df[df['mass'].i-
sNull()].s- snull()).head()
how(5)
```

```
df[(df['mass']<- df[(df['mass']<-
3400) & (df['s- 3400) & (df['s-
ex']=='Ma- ex']=='Ma-
le')].head() le')].show(5)
```

```
df[~df['flip- df[~df['flip-
per'].betwee- per'].betwee-
n(225,229)].s- n(225,229)].h-
how(5) ead()
```

Conditional Transformations

```
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')
```



By **datamansam**

Published 3rd May, 2022.

Last updated 28th May, 2022.

Page 1 of 1.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>