Pandas-1 In [1]: #importing pandas import pandas as pd In [3]: #Read csv file and store in variable data data=pd.read_csv("addresses.csv") In [4]: **#To display data** data id location_id state_province postal_code country address_1 address_2 Out[4]: 2600 Middlefield Road 0 1 1 NaN Redwood City CA 94063 US US 1 2 2 24 Second Avenue NaN San Mateo CA 94401 2 3 3 24 Second Avenue NaN San Mateo CA 94403 US US 3 4 4 24 Second Avenue CA 94401 NaN San Mateo 4 5 5 24 Second Avenue San Mateo CA 94401 US NaN 6 6 800 Middle Avenue NaN Menlo Park CA 94025-9881 US 6 7 7 500 Arbor Road NaN Menlo Park CA 94025 US 8 94025-9881 8 800 Middle Avenue CA US NaN Menlo Park 8 9 9 2510 Middlefield Road NaN CA 94063 US Redwood City 9 10 10 1044 Middlefield Road 94063 US NaN Redwood City CA **10** 11 11 2140 Euclid Avenue. NaN 94061 US Redwood City CA **11** 12 12 1044 Middlefield Road Redwood City CA 94063 US 2nd Floor **12** 13 13 399 Marine Parkway. Redwood City 94065 US NaN CA **13** 14 14 660 Veterans Blvd. NaN Redwood City CA 94063 US San Francisco **14** 15 15 1500 Valencia Street 94110 US NaN CA 16 94087 US 15 16 1161 South Bernardo NaN Sunnyvale CA **16** 17 17 409 South Spruce Avenue NaN South San Francisco CA 94080 US 18 114 Fifth Avenue Redwood City 94063 US **17** 18 NaN CA **18** 19 19 19 West 39th Avenue San Mateo CA 94403 US NaN US **19** 20 21 123 El Camino Real CA 94002 NaN Belmont **20** 21 22 2013 Avenue of the fellows Suite 100 San Francisco 94103 US CA In [5]: #Display top 5 data rows in dataset data.head() id location_id Out[5]: address_1 address_2 city state_province postal_code country 0 1 1 2600 Middlefield Road 94063 US NaN Redwood City CA **1** 2 CA 94401 US 24 Second Avenue NaN San Mateo 2 3 3 24 Second Avenue NaN San Mateo 94403 US CA 24 Second Avenue 94401 4 CA US **3** 4 NaN San Mateo 24 Second Avenue 5 CA 94401 US 4 5 NaN San Mateo In [6]: #Display last 5 rows in dataset data.tail() Out[6]: id location_id address_1 address_2 city state_province postal_code country **16** 17 17 409 South Spruce Avenue NaN South San Francisco 94080 US CA **17** 18 18 114 Fifth Avenue NaN Redwood City CA 94063 US US **18** 19 19 94403 19 West 39th Avenue NaN San Mateo CA 123 El Camino Real 19 20 21 NaN **Belmont** CA 94002 US **20** 21 94103 US 22 2013 Avenue of the fellows Suite 100 San Francisco CA In [9]: **#Give** count of record data.count() id 21 Out[9]: location_id 21 address_1 21 address_2 2 21 city 21 state_province postal_code 21 country 21 dtype: int64 In [10]: #Checking null value data.isnull() id location_id address_1 address_2 city state_province postal_code country Out[10]: o False False False True False False False False 1 False False False True False False False False 2 False False False True False False False False **3** False False False True False False False False 4 False False False True False False False False **5** False False False True False False False False 6 False False False True False False False False **7** False False False False True False False False 8 False False False True False False False False 9 False False False True False False False False 10 False False False True False False False False 11 False False False False False False False False 12 False False True False False False False False 13 False False False True False False False False 14 False False False True False False False False **15** False False False True False False False False 16 False False False False True False False False 17 False False False True False False False False 18 False False False False False True False False False False False 19 False True False False False 20 False False False False False False False False In [11]: **#Gives** statastical values data.describe() Out[11]: id location_id count 21.000000 21.000000 mean 11.000000 11.095238 6.204837 std 6.363213 min 1.000000 1.000000 **25**% 6.000000 6.000000 **50%** 11.000000 11.000000 **75%** 16.000000 16.000000 max 21.000000 22.000000 In [13]: #Gievs type of dataset type(data) pandas.core.frame.DataFrame Out[13]: In [14]: **#Gives all columns names** data.columns Index(['id', 'location_id', 'address_1', 'address_2', 'city', 'state_province', Out[14]: 'postal_code', 'country'], dtype='object') In [15]: list(data.columns) ['id', Out[15]: 'location_id', 'address_1', 'address_2', 'city', 'state_province', 'postal_code', 'country'] In [17]: #Display city column data['city'] Redwood City Out[17]: San Mateo 2 San Mateo 3 San Mateo San Mateo 4 Menlo Park 5 6 Menlo Park 7 Menlo Park 8 Redwood City 9 Redwood City 10 Redwood City 11 Redwood City 12 Redwood City 13 Redwood City 14 San Francisco 15 Sunnyvale 16 South San Francisco 17 Redwood City 18 San Mateo 19 Belmont 20 San Francisco Name: city, dtype: object In [18]: #Display city column with column name data[['city']] Out[18]: city 0 Redwood City 1 San Mateo 2 San Mateo 3 San Mateo 4 San Mateo 5 Menlo Park 6 Menlo Park 7 Menlo Park 8 Redwood City 9 Redwood City 10 Redwood City Redwood City 11 12 Redwood City 13 Redwood City 14 San Francisco 15 Sunnyvale South San Francisco 16 17 Redwood City 18 San Mateo 19 Belmont 20 San Francisco In [19]: #Display type of given column type(data[['city']]) pandas.core.frame.DataFrame Out[19]: In [22]: **#Displays multiple columns** data[['city', 'address_1']] Out[22]: city address_1 0 Redwood City 2600 Middlefield Road 24 Second Avenue 1 San Mateo 2 San Mateo 24 Second Avenue 3 San Mateo 24 Second Avenue 4 San Mateo 24 Second Avenue 800 Middle Avenue 5 Menlo Park 6 Menlo Park 500 Arbor Road 7 800 Middle Avenue Menlo Park 2510 Middlefield Road 8 Redwood City 9 Redwood City 1044 Middlefield Road 2140 Euclid Avenue. 10 Redwood City 11 Redwood City 1044 Middlefield Road 12 Redwood City 399 Marine Parkway. 660 Veterans Blvd. 13 Redwood City 14 San Francisco 1500 Valencia Street 1161 South Bernardo 15 Sunnyvale South San Francisco 16 409 South Spruce Avenue 17 Redwood City 114 Fifth Avenue 19 West 39th Avenue 18 San Mateo 123 El Camino Real Belmont 20 San Francisco 2013 Avenue of the fellows In [25]: #To read excel,xls type of file df=pd.read_excel('airline.xls') In [26]: **#Display** dataset YEAR Υ W R L Κ Out[26]: **0** 1948 1.214 0.243 0.1454 1.415 0.612 1 1949 1.354 0.260 0.2181 1.384 0.559 1950 1.569 0.278 0.3157 1.388 0.573 1951 1.948 0.297 0.3940 1.550 0.564 1952 2.265 0.310 0.3559 1.802 0.574 4 1953 2.731 0.322 0.3593 1.926 0.711 3.025 0.335 0.4025 1.964 0.776 6 1954 1955 3.562 0.350 0.3961 2.116 0.827 3.979 0.361 0.3822 2.435 0.800 8 1956 1957 4.420 0.379 0.3045 2.707 0.921 4.563 0.391 0.3284 2.706 1.067 10 1958 5.385 0.426 0.3856 2.846 1.083 11 1959 12 1960 5.554 0.441 0.3193 3.089 1.481 13 1961 5.465 0.460 0.3079 3.122 1.736 5.825 0.485 0.3783 3.184 1.926 14 1962 6.876 0.506 0.4180 3.263 2.041 1963 15 16 1964 7.823 0.538 0.5163 3.412 1.997 17 1965 9.120 0.564 0.5879 3.623 2.257 1966 10.512 0.586 0.5369 4.074 2.742 18 1967 13.020 0.622 0.4443 4.710 3.564 19 1968 15.261 0.666 0.3052 5.217 4.767 20 21 1969 16.313 0.731 0.2332 5.569 6.511 1970 16.002 0.831 0.1883 5.495 7.627 1971 15.876 0.906 0.2023 5.334 8.673 23 1972 16.662 1.000 0.2506 5.345 8.331 24 1973 17.014 1.056 0.2668 5.662 8.557 25 1974 19.305 1.131 0.2664 5.729 9.508 26 1975 18.721 1.247 0.2301 5.722 9.062 27 1976 19.250 1.375 0.3452 5.762 8.262 28 1977 20.647 1.544 0.4508 5.877 7.474 29 1978 22.726 1.703 0.5877 6.108 7.104 30 31 1979 23.619 1.779 0.5346 6.852 6.874 In [27]: df.head() Out[27]: Υ W Κ YEAR R L 1948 1.214 0.243 0.1454 1.415 0.612 1949 1.354 0.260 0.2181 1.384 0.559 1950 1.569 0.278 0.3157 1.388 0.573 1951 1.948 0.297 0.3940 1.550 0.564 1952 2.265 0.310 0.3559 1.802 0.574 In [28]: df1=pd.read_excel('airline.xls') In [29]: df1 Out[29]: YEAR Υ W R L Κ 1948 1.214 0.243 0.1454 1.415 0.612 1949 1.354 0.260 0.2181 1.384 0.559 1950 1.569 0.278 0.3157 1.388 0.573 1951 1.948 0.297 0.3940 1.550 0.564 1952 2.265 0.310 0.3559 1.802 0.574 1953 2.731 0.322 0.3593 1.926 0.711 1954 3.025 0.335 0.4025 1.964 0.776 6 1955 3.562 0.350 0.3961 2.116 0.827 8 1956 3.979 0.361 0.3822 2.435 0.800 1957 4.420 0.379 0.3045 2.707 0.921 10 1958 4.563 0.391 0.3284 2.706 1.067 1959 5.385 0.426 0.3856 2.846 1.083 11 12 1960 5.554 0.441 0.3193 3.089 1.481 13 1961 5.465 0.460 0.3079 3.122 1.736 1962 5.825 0.485 0.3783 3.184 1.926 14 **15** 1963 6.876 0.506 0.4180 3.263 2.041 7.823 0.538 0.5163 3.412 1.997 16 1964 9.120 0.564 0.5879 3.623 2.257 17 1965 1966 10.512 0.586 0.5369 4.074 2.742 18 1967 13.020 0.622 0.4443 4.710 3.564 19 1968 15.261 0.666 0.3052 5.217 4.767 20 1969 16.313 0.731 0.2332 5.569 6.511 21 22 1970 16.002 0.831 0.1883 5.495 7.627 23 1971 15.876 0.906 0.2023 5.334 8.673 1972 16.662 1.000 0.2506 5.345 8.331 1973 17.014 1.056 0.2668 5.662 8.557 1974 19.305 1.131 0.2664 5.729 9.508 26 1975 18.721 1.247 0.2301 5.722 9.062 1976 19.250 1.375 0.3452 5.762 8.262 28 29 1977 20.647 1.544 0.4508 5.877 7.474 1978 22.726 1.703 0.5877 6.108 7.104 1979 23.619 1.779 0.5346 6.852 6.874 In [32]: #Reading sheet 2 from airline1.xls pd.read_excel("airline1.xls" , sheet_name = "Sheet2") YEAR Υ W R K Out[32]: 1948 1.214 0.243 0.1454 1.415 0.612 0 **1** 1949 1.354 0.260 0.2181 1.384 0.559 1950 1.569 0.278 0.3157 1.388 0.573 1951 1.948 0.297 0.3940 1.550 0.564 1952 2.265 0.310 0.3559 1.802 0.574 1953 2.731 0.322 0.3593 1.926 0.711 1954 3.025 0.335 0.4025 1.964 0.776 1955 3.562 0.350 0.3961 2.116 0.827 1956 3.979 0.361 0.3822 2.435 0.800 In [35]: #Reading sheet3 from airline1 and giving column name to sheet dt = pd.read_excel("airline1.xls" , sheet_name = "Sheet3" , header=None , names=['a', 'b' , 'c','d','e','f']) In [36]: dt b d f Out[36]: С е **0** 1960 5.554 0.441 0.3193 3.089 1.481 **1** 1961 5.465 0.460 0.3079 3.122 1.736 **2** 1962 5.825 0.485 0.3783 3.184 1.926 **3** 1963 6.876 0.506 0.4180 3.263 2.041 7.823 0.538 0.5163 3.412 1.997 **4** 1964 9.120 0.564 0.5879 3.623 2.257 **5** 1965 **6** 1966 10.512 0.586 0.5369 4.074 2.742 **7** 1967 13.020 0.622 0.4443 4.710 3.564 **8** 1968 15.261 0.666 0.3052 5.217 4.767 **9** 1969 16.313 0.731 0.2332 5.569 6.511 **10** 1970 16.002 0.831 0.1883 5.495 7.627 **11** 1971 15.876 0.906 0.2023 5.334 8.673 **12** 1972 16.662 1.000 0.2506 5.345 8.331 **13** 1973 17.014 1.056 0.2668 5.662 8.557 **14** 1974 19.305 1.131 0.2664 5.729 9.508 **15** 1975 18.721 1.247 0.2301 5.722 9.062 In [37]: dt.to_csv('test1.csv' , index= False) In [38]: dt.to_excel("test2.xlsx") In [39]: dt.head() Out[39]: b С d f е **0** 1960 5.554 0.441 0.3193 3.089 1.481 **1** 1961 5.465 0.460 0.3079 3.122 1.736 **2** 1962 5.825 0.485 0.3783 3.184 1.926 **3** 1963 6.876 0.506 0.4180 3.263 2.041 **4** 1964 7.823 0.538 0.5163 3.412 1.997 In [40]: dt.tail() Out[40]: **11** 1971 15.876 0.906 0.2023 5.334 8.673 **12** 1972 16.662 1.000 0.2506 5.345 8.331 **13** 1973 17.014 1.056 0.2668 5.662 8.557 **14** 1974 19.305 1.131 0.2664 5.729 9.508 **15** 1975 18.721 1.247 0.2301 5.722 9.062 In [41]: type(dt) pandas.core.frame.DataFrame Out[41]: In [42]: dt.columns Index(['a', 'b', 'c', 'd', 'e', 'f'], dtype='object') Out[42]: In [43]: list(dt.columns) ['a', 'b', 'c', 'd', 'e', 'f'] Out[43]: In [44]: dt[['a','d']] Out[44]: **0** 1960 0.3193 **1** 1961 0.3079 **2** 1962 0.3783 **3** 1963 0.4180 **4** 1964 0.5163 **5** 1965 0.5879 **6** 1966 0.5369 **7** 1967 0.4443 **8** 1968 0.3052 **9** 1969 0.2332 **10** 1970 0.1883 **11** 1971 0.2023 **12** 1972 0.2506 **13** 1973 0.2668 **14** 1974 0.2664 **15** 1975 0.2301 In [47]: d=pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv') In [48]: Fare Cabin Embarked Out[48]: Passengerld Survived Pclass Sex Age SibSp Parch Ticket Name male 22.0 1 Braund, Mr. Owen Harris A/5 21171 7.2500 NaN Cumings, Mrs. John Bradley (Florence Briggs PC 17599 71.2833 female 38.0 C85 STON/O2. S 2 3 1 3 7.9250 Heikkinen, Miss. Laina female 26.0 NaN 3101282 53.1000 S 3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 113803 C123 4 5 0 3 Allen, Mr. William Henry 0 0 373450 8.0500 S male 35.0 NaN 886 887 0 2 0 0 211536 13.0000 NaN S Montvila, Rev. Juozas male 27.0 887 888 Graham, Miss. Margaret Edith female 19.0 112053 30.0000 B42 S Johnston, Miss. Catherine Helen "Carrie" 888 889 0 3 2 W./C. 6607 23.4500 S female NaN 1 NaN Behr, Mr. Karl Howell С 889 890 male 26.0 111369 30.0000 C148 0 3 male 32.0 Q 890 891 0 0 370376 7.7500 Dooley, Mr. Patrick NaN 891 rows × 12 columns In [49]: d.head() Cabin Embarked Passengerld Survived Pclass Name Sex Age SibSp Parch **Ticket Fare** Out[49]: 0 1 0 3 Braund, Mr. Owen Harris male 22.0 1 0 A/5 21171 7.2500 NaN S Cumings, Mrs. John Bradley (Florence Briggs female 38.0 1 0 PC 17599 71.2833 C85 С STON/O2. 2 3 1 S 3 Heikkinen, Miss. Laina female 26.0 0 0 7.9250 NaN 3101282 Futrelle, Mrs. Jacques Heath (Lily May Peel) 3 1 female 35.0 1 0 113803 53.1000 C123 S 5 0 3 Allen, Mr. William Henry 0 0 S 4 373450 8.0500 male 35.0 NaN In [50]: #Rading html file x=pd.read_html("https://www.basketball-reference.com/leagues/NBA_2015_totals.html") In [51]: Rk Player Pos Age Tm G GS MP FG FGA FT% 0RB Out[51]: 1 Quincy Acy ΡF 24 NYK 68 22 1287 152 331 .784 79 . . . 1 2 Jordan Adams SG 20 MEM30 0 248 35 86 .609 9 . . . 2 3 Steven Adams С 21 0KC 70 67 1771 217 399 .502 199 3 ΡF 215 4 Jeff Adrien 28 MIN17 0 19 44 .579 23 72 2502 4 5 Arron Afflalo SG 29 T0T 78 375 884 .843 27 ΡF TOT 670 490 Thaddeus Young 26 76 68 2434 451 968 . 655 127 Thaddeus Young ΡF MIN1605 289 671 490 26 48 48 641 .682 75 672 490 Thaddeus Young ΡF 26 BRK 28 20 829 162 327 .606 52 С CH0 673 491 Cody Zeller 22 62 45 1487 172 373 .774 97 492 Tyler Zeller С 25 B0S 82 59 340 674 1731 619 .823 146 DRB TRB AST STL BLK TOV ΡF PTS 0 222 301 68 27 22 60 147 398 1 19 28 16 16 7 14 24 94 2 324 523 66 38 99 222 537 86 3 54 15 9 9 77 4 30 60 4 220 247 129 41 7 116 167 1035 670 284 411 173 124 25 117 171 1071 170 245 135 86 17 75 115 685 671 8 56 672 114 166 38 38 42 386 265 362 100 34 49 62 156 472 673 319 465 113 18 52 76 205 833 [675 rows \times 30 columns]] In [52]: type(x) list Out[52]: In [53]: len(x) Out[53]: In [54]: #Diplay in tabuler format x[0] MP FG FGA ... FT% ORB DRB TRB AST STL BLK TOV Out[54]: Rk Player Pos Age Tm G GS PF **PTS** 0 1 Quincy Acy 24 NYK 68 22 1287 152 331784 79 222 301 68 27 22 60 147 398 Jordan Adams SG 86609 28 20 MEM 30 0 248 35 9 19 16 16 7 14 24 94 Steven Adams 2 С 523 3 21 OKC 70 67 1771 217 399502 199 324 66 38 86 99 222 537 3 Jeff Adrien PF 0 215 44579 77 15 9 9 30 60 28 MIN 17 19 23 54 4 TOT 78 72 2502 375 884843 27 220 247 129 41 7 116 167 1035 5 Arron Afflalo SG 29 670 490 Thaddeus Young TOT 76 68 2434 451 968 ... 127 284 411 124 25 117 171 1071 26 .655 173 **671** 490 Thaddeus Young 17 75 115 PF 26 MIN 48 48 1605 289 641682 75 170 245 135 86 685 672 490 Thaddeus Young 26 BRK 28 20 829 162 327606 52 114 166 38 38 42 56 386 97 **673** 491 Cody Zeller 22 CHO 62 45 1487 172 373774 362 49 62 156 С 265 100 34 472 **674** 492 С 619823 146 465 Tyler Zeller 25 BOS 82 59 1731 340 319 113 18 52 76 205 833 675 rows × 30 columns In []: