We've accomplished a lot so far in this course and now it's time to put our data cleaning skills to work!

Working on *guided projects* gives you hands on experience with real world examples, which also means they'll be more challenging than missions. However, keep in mind that now you have more tools you can use to clean and transform data than you did at the beginning of this course, including:

* Vectorized string methods to clean string columns
* The apply(), map(), and applymap() methods to transform data
* The fillna(), dropna(), and drop() methods to drop missing or unnecessary values
* The melt() function to reshape data
* The concat() and merge() functions to combine data

If you get stuck, don't be afraid to:

* Review previous missions.
* Use Google and StackOverflow to search for answers.

It's very difficult to understand every concept and memorize syntax the first time you complete a course, so don't be discouraged if you need to review missions again or use Google to search for answers.

In this guided project, we'll work with exit surveys from employees of the [Department of Education, Training and Employment](https://en.wikipedia.org/wiki/Department_of_Education_and_Training_(Queensland)) (DETE) and the Technical and Further Education (TAFE) institute in Queensland, Australia. You can find the TAFE exit survey [here](https://data.gov.au/dataset/ds-qld-89970a3b-182b-41ea-aea2-6f9f17b5907e/details?q=exit%20survey) and the survey for the DETE [here](https://data.gov.au/dataset/ds-qld-fe96ff30-d157-4a81-851d-215f2a0fe26d/details?q=exit%20survey). We've made some slight modifications to these datasets to make them easier to work with, including changing the encoding to UTF-8 (the original ones are encoded using cp1252.)

In this project, we'll play the role of data analyst and pretend our stakeholders want to know the following:

* Are employees who only worked for the institutes for a short period of time resigning due to some kind of dissatisfaction? What about employees who have been there longer?
* Are younger employees resigning due to some kind of dissatisfaction? What about older employees?

They want us to combine the results for *both* surveys to answer these questions. However, although both used the same survey template, one of them customized some of the answers. In the guided steps, we'll aim to do most of the data cleaning and get you started analyzing the first question.

A data dictionary wasn't provided with the dataset. In a job setting, we'd make sure to meet with a manager to confirm the definitions of the data. For this project, we'll use our general knowledge to define the columns.

Below is a preview of a couple columns we'll work with from the dete\_survey.csv:

* ID: An id used to identify the participant of the survey
* SeparationType: The reason why the person's employment ended
* Cease Date: The year or month the person's employment ended
* DETE Start Date: The year the person began employment with the DETE

Below is a preview of a couple columns we'll work with from the tafe\_survey.csv:

* Record ID: An id used to identify the participant of the survey
* Reason for ceasing employment: The reason why the person's employment ended
* LengthofServiceOverall. Overall Length of Service at Institute (in years): The length of the person's employment (in years)

Let's start by reading the datasets into pandas and exploring them.

Instructions

* Start by writing a paragraph in a markdown cell introducing the project and the dataset.
* Import the pandas and NumPy libraries.
* Read the dete\_survey.csv CSV file into pandas, and assign it to the variable name dete\_survey.
* Read the tafe\_survey.csv CSV file into pandas, and assign it to the variable name tafe\_survey.
* Use the [DataFrame.info()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.info.html) and [DataFrame.head()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.head.html" \t "_blank) methods to print information about both dataframes, as well as the first few rows. Use other data exploration methods such as the [Series.value\_counts()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.value_counts.html" \t "_blank) and [DataFrame.isnull()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isnull.html" \t "_blank) methods to explore the data and figure out some next steps.
  + Write a markdown cell briefly describing your observations.

Often, figuring out the steps you need to take to clean and reshape your data is the hardest part. If you couldn't find a clear path forward in the last screen, don't worry! We'll lay out the steps for you, but give you some room to start making your own decisions.

From our work in the previous screen, we can first make the following observations:

* The dete\_survey dataframe contains 'Not Stated' values that indicate values are missing, but they aren't represented as NaN.
* Both the dete\_survey and tafe\_survey dataframes contain many columns that we don't need to complete our analysis.
* Each dataframe contains many of the same columns, but the column names are different.
* There are multiple columns/answers that indicate an employee resigned because they were dissatisfied.

To start, we'll handle the first two issues. Recall that we can use the [pd.read\_csv()function](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html" \t "_blank) to specify values that should be represented as NaN. We'll use this function to fix the missing values first. Then, we'll drop columns we know we don't need for our analysis.

Instructions

* Read the dete\_survey.csv CSV file into pandas again, but this time read the Not Stated values in as NaN.
  + To read Not Stated in as NaN, set the na\_values parameter to Not Statedin the [pd.read\_csv()](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html" \t "_blank) function.
  + Assign the result to the variable name dete\_survey.
* Then, let's drop some columns from each dataframe that we won't use in our analysis to make the dataframes easier to work with.
  + Use the [DataFrame.drop() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html" \t "_blank) to drop the following columns from dete\_survey: dete\_survey.columns[28:49]. Remember to set the axisparameter equal to 1.
    - Assign the result to dete\_survey\_updated.
  + Use the DataFrame.drop() method to drop the following columns from tafe\_survey: tafe\_survey.columns[17:66]. Remember to set the axisparameter equal to 1.
    - Assign the result to tafe\_survey\_updated.
* Write a markdown cell explaining the changes you made and why.

Next, let's turn our attention to the column names. Each dataframe contains many of the same columns, but the column names are different. Below are some of the columns we'd like to use for our final analysis:

|  |  |  |
| --- | --- | --- |
| dete\_survey | tafe\_survey | Definition |
| ID | Record ID | An id used to identify the participant of the survey |
| SeparationType | Reason for ceasing  employment | The reason why the participant's employment ended |
| Cease Date | CESSATION YEAR | The year or month the participant's employment ended |
| DETE Start Date |  | The year the participant began employment with the DETE |
|  | LengthofServiceOverall.  Overall Length of Service  at Institute (in years) | The length of the person's employment (in years) |
| Age | CurrentAge.  Current Age | The age of the participant |
| Gender | Gender.  What is your Gender? | The gender of the participant |

Because we eventually want to combine them, we'll have to standardize the column names. Recall that we can use the [DataFrame.columns](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.columns.html" \t "_blank) attribute along with vectorized string methods to update all of the columns at once. Here's an example from the last mission:

happiness2017.columns = happiness2017.columns.str.replace('.', ' ').str.replace('\s+', ' ').str.strip().str.upper()

Instructions

* Rename the remaining columns in the dete\_survey\_updated dataframe.
  + Use the following criteria to update the column names:
    - Make all the capitalization lowercase.
    - Remove any trailing whitespace from the end of the strings.
    - Replace spaces with underscores ('\_').
  + As an example, Cease Date should be updated to cease\_date.
  + Remember you can use the [DataFrame.columns](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.columns.html" \t "_blank) attribute to print an array of the existing column names.
* Use the [DataFrame.rename() method](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.rename.html" \t "_blank) to update the columns below in tafe\_survey\_updated. Don't worry about the rest of the column names right now - we'll handle them later.
  + 'Record ID': 'id'
  + 'CESSATION YEAR': 'cease\_date'
  + 'Reason for ceasing employment': 'separationtype'
  + 'Gender. What is your Gender?': 'gender'
  + 'CurrentAge. Current Age': 'age'
  + 'Employment Type. Employment Type': 'employment\_status'
  + 'Classification. Classification': 'position'
  + 'LengthofServiceOverall. Overall Length of Service at Institute (in years)': 'institute\_service'
  + 'LengthofServiceCurrent. Length of Service at current workplace (in years)': 'role\_service'
* Use the [DataFrame.head() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.head.html" \t "_blank) to look at the current state of the dete\_survey\_updated and tafe\_survey\_updated dataframes and make sure your changes look good.
* Write a markdown cell explaining the changes you made and why.

In the last screen, we renamed the columns that we'll use in our analysis. Next, let's remove more of the data we don't need.

Recall that our end goal is to answer the following question:

* Are employees who have only worked for the institutes for a short period of time resigning due to some kind of dissatisfaction? What about employees who have been at the job longer?

If we look at the unique values in the separationtype columns in each dataframe, we'll see that each contains a couple of different separation types. For this project, we'll only analyze survey respondents who *resigned*, so their separation type contains the string 'Resignation'.

If you're interested in a challenge, try to complete the project using *all* of the separation types instead - you'll find more issues to work through in the data cleaning process.

Note that dete\_survey\_updated dataframe contains multiple separation types with the string 'Resignation':

* Resignation-Other reasons
* Resignation-Other employer
* Resignation-Move overseas/interstate

Remember that we'll have to account for each of these variations so we don't unintentionally drop data!

In this step, note that you may see what is known as a [SettingWithCopy Warning](https://www.dataquest.io/blog/settingwithcopywarning/" \t "_blank). This won't prevent your code from running properly but it's just letting you know that whatever operation you're doing is trying to be set on a copy of a slice from a dataframe. We'll include instructions below to get around this.

Instructions

* Use the [Series.value\_counts() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.value_counts.html" \t "_blank) to review the unique values in the separationtype column in both dete\_survey\_updated and tafe\_survey\_updated.
* In each of dataframes, select only the data for survey respondents who have a Resignation separation type.
  + Remember that the dete\_survey\_updated dataframe contains three Resignation separation types. We want to select all of them.
  + Use the [DataFrame.copy() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.copy.html" \t "_blank) on the result to avoid the SettingWithCopy Warning.
  + Assign the result for dete\_survey\_updated to dete\_resignations.
  + Assign the result for tafe\_survey\_updated to tafe\_resignations.
* Write a markdown paragraph explaining the changes you made and why.

Now, before we start cleaning and manipulating the rest of our data, let's verify that the data doesn't contain any major inconsistencies (to the best of our knowledge). When you're working with real world data, don't assume that the data you're analyzing isn't corrupted in some way!

It may not always be possible to catch all of these errors, but by making sure the data seems reasonable to the best of our knowledge, we can stop ourselves from completing a data analysis project that winds up being useless because of bad data.

In this step, we'll focus on verifying that the years in the cease\_date and dete\_start\_date columns make sense. However, we encourage you to check the data for other issues as well!

* Since the cease\_date is the last year of the person's employment and the dete\_start\_date is the person's first year of employment, it wouldn't make sense to have years after the current date.
* Given that most people in this field start working in their 20s, it's also unlikely that the dete\_start\_date was before the year 1940.

If we have many years higher than the current date or lower than 1940, we wouldn't want to continue with our analysis, because it could mean there's something very wrong with the data. If there are a small amount of values that are unrealistically high or low, we can remove them.

Instructions

* Check the years in each dataframe for logical inconsistencies.
  + First, clean the cease\_date column in dete\_resignations.
    - Use the Series.value\_counts() method to view the unique values in the cease\_date column.
    - Use vectorized string methods to extract the year. As a reminder, [here](https://pandas.pydata.org/pandas-docs/stable/text.html#text-summary) is the full list.
    - Use the [Series.astype() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.astype.html" \t "_blank) method to convert the type to a float.
  + Use the Series.value\_counts() to check the values in the cease\_date and dete\_start\_date columns in dete\_resignations and the cease\_datecolumn in tafe\_resignations.
    - Because Series.value\_counts() returns a series, we can use [Series.sort\_index() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.sort_index.html" \t "_blank) with ascending= True or False to view the highest and lowest values with their counts.
  + You can also plot the values of any numeric columns with [a boxplot](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.boxplot.html) to identify any values that look wrong.
* Write a markdown paragraph explaining your findings.

From the work we did in the last screen, we can verify:

1. There aren't any major issues with the years.
2. The years in each dataframe don't span quite the same number of years. We'll leave it up to your discretion to drop any years you don't think are needed for the analysis.

Now that we've verified the years in the dete\_resignations dataframe, we'll use them to create a new column. Recall that our end goal is to answer the following question:

* Are employees who have only worked for the institutes for a short period of time resigning due to some kind of dissatisfaction? What about employees who have been at the job longer?

In the Human Resources field, the length of time an employee spent in a workplace is referred to as their years of *service*.

You may have noticed that the tafe\_resignations dataframe already contains a "service" column, which we renamed to institute\_service. In order to analyze both surveys together, we'll have to create a corresponding institute\_service column in dete\_resignations.

Do we have data that can be used to calculate the length of time the employee spent in their workplace? Take a minute to review dete\_resignations once more and see if you can answer this question before moving on.

Instructions

* Create an institute\_service column in dete\_resignations
  + Create a new column named institute\_service in dete\_resignations.
    - Subtract the dete\_start\_date from the cease\_date. Assign the result to a new column named institute\_service.
* Write a markdown paragraph explaining the changes you made and why.

In the last screen, we created a new institute\_service column that we'll use to analyze survey respondents according to their length of employment.Next, we'll identify any employees who resigned because they were dissatisfied.

Below are the columns we'll use to categorize employees as "dissatisfied" from each dataframe. If you disagree, feel free to modify them! Just make sure you explain *why* you made that decision.

1. tafe\_survey\_updated:
   * Contributing Factors. Dissatisfaction
   * Contributing Factors. Job Dissatisfaction
2. dete\_survey\_updated:
   * job\_dissatisfaction
   * dissatisfaction\_with\_the\_department
   * physical\_work\_environment
   * lack\_of\_recognition
   * lack\_of\_job\_security
   * work\_location
   * employment\_conditions
   * work\_life\_balance
   * workload

If the employee indicated any of the factors above caused them to resign, we'll mark them as dissatisfied in a new column.

To create the new column, we'll do the following:

1. Convert the values in the 'Contributing Factors. Dissatisfaction' and 'Contributing Factors. Job Dissatisfaction' columns in the tafe\_resignations dataframe to True, False, or NaN values.
2. If any of the columns listed above contain a True value, we'll add a True value to a new column named dissatisfied. To accomplish this, we'll use the [DataFrame.any() method](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.any.html" \t "_blank) to do the following:
   * Return True if *any* element in the selected columns above is True
   * Return False if *none* of the elements in the selected columns above is True
   * Return NaN if the value is NaN

Here's the syntax we can use:

df.any(axis=1, skipna=False)

After our changes, the new dissatisfied column will contain just the following values:

* True: indicates a person resigned because they were dissatisfied with the job
* False: indicates a person resigned because of a reason other than dissatisfaction with the job
* NaN: indicates the value is missing

Instructions

* Use the Series.value\_counts() method to view the values in the 'Contributing Factors. Dissatisfaction' and 'Contributing Factors. Job Dissatisfaction' in the tafe\_resignations dataframe.
* Update the values in the 'Contributing Factors. Dissatisfaction' and 'Contributing Factors. Job Dissatisfaction' in the tafe\_resignationsdataframe so that each contains only True, False, or NaN values.
  + Write a function named update\_vals that makes the following changes:
    - If the value is NaN, return np.nan. You can use the following criteria to check that a value is NaN: pd.isnull(val).
    - If the value is '-', return False.
    - For any other value, return True.
  + Use the [DataFrame.applymap() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.applymap.html" \t "_blank) to apply the function above to the 'Contributing Factors. Dissatisfaction' and 'Contributing Factors. Job Dissatisfaction' in the tafe\_resignations dataframe.
    - Remember that we need to pass the update\_vals function into the df.applymap() method *without* parentheses.
* Use the df.any() method as described above to create a dissatisfied column in BOTH the tafe\_resignations and dete\_resignations dataframes.
* Use the df.copy() method to create a copy of the results and avoid the SettingWithCopy Warning. Assign the results to dete\_resignations\_up and tafe\_resignations\_up.
* Write a markdown paragraph explaining the changes you made and why.

To recap, we've accomplished the following:

* Renamed our columns
* Dropped any data not needed for our analysis
* Verified the quality of our data
* Created a new institute\_service column
* Cleaned the Contributing Factors columns
* Created a new column indicating if an employee resigned because they were dissatisfied in some way

Now, we're finally ready to combine our datasets! Our end goal is to aggregate the data according to the institute\_service column, so when you combine the data, think about how to get the data into a form that's easy to aggregate.

Instructions

* First, let's add a column to each dataframe that will allow us to easily distinguish between the two.
  + Add a column named institute to dete\_resignations\_up. Each row should contain the value DETE.
  + Add a column named institute to tafe\_resignations\_up. Each row should contain the value TAFE.
* Combine the dataframes. Assign the result to combined.
* Recall that we still have some columns left in the dataframe that we don't need to complete our analysis. Use the [DataFrame.dropna() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.dropna.html" \t "_blank) to drop any columns with less than 500 non null values.
  + Remember that you can drop columns with less than a certain number of non null values with the thresh parameter.
  + Assign the result to combined\_updated.
* Write a markdown paragraph explaining the changes you made and why.

Now that we've combined our dataframes, we're almost at a place where we can perform some kind of analysis! First, though, we'll have to clean up the institute\_servicecolumn. This column is tricky to clean because it currently contains values in a couple different forms:

NaN                 88

Less than 1 year   73

1-2                 64

3-4                 63

5-6                 33

11-20               26

5.0                 23

1.0                 22

7-10               21

0.0                 20

...

To analyze the data, we'll convert these numbers into categories. We'll base our analysis on [this article](https://www.businesswire.com/news/home/20171108006002/en/Age-Number-Engage-Employees-Career-Stage), which makes the argument that understanding employee's needs according to career stage instead of age is more effective.

We'll use the slightly modified definitions below:

* New: Less than 3 years at a company
* Experienced: 3-6 years at a company
* Established: 7-10 years at a company
* Veteran: 11 or more years at a company

Let's categorize the values in the institute\_service column using the definitions above.

Instructions

* First, we'll extract the years of service from each value in the institute\_servicecolumn.
  + Use the [Series.astype() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.astype.html" \t "_blank) to change the type to 'str'.
  + Use vectorized string methods to extract the years of service from each pattern. You can find the full list of vectorized string methods [here](https://pandas.pydata.org/pandas-docs/stable/text.html#text-summary).
  + Double check that you didn't miss extracting any digits.
  + Use the Series.astype() method to change the type to 'float'.
* Next, we'll map each value to one of the career stage definitions above.
  + Create a function that maps each year value to one of the career stages above.
    - Remember that you'll have to handle missing values separately. You can use the following code to check if a value is NaN where val is the name of the value: pd.isnull(val).
  + Use the [Series.apply() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.apply.html" \t "_blank) to apply the function to the institute\_service column. Assign the result to a new column named service\_cat.
* Write a markdown paragraph explaining the changes you made and why.

In the last screen, we created a service\_cat column, that categorizes employees according to the amount of years spent in their workplace:

* New: Less than 3 years at a company
* Experienced: 3-6 years at a company
* Established: 7-10 years at a company
* Veteran: 11 or more years at a company

Now, let's finally do our first piece of analysis! We'll help you fill in missing values in the dissatisfied column and then aggregate the data to get you started, but note that we still have additional missing values left to deal with. This is meant to be an initial introduction to the analysis, *not* the final analysis.

Recall that the dissatisfied column consists of Boolean values, meaning they're either True or False. Methods such as the df.pivot\_table() method actually treat Boolean values as integers, so a True value is considered to be 1 and a False value is considered to be 0. That means that we can aggregate the dissatisfied column and calculate the number of people in each group, the percentage of people in each group, etc.

Instructions

* Use the Series.value\_counts() method to confirm if the number of True and False in the dissatisfied column. Set the dropna parameter to False to also confirm the number of missing values.
* Use the [DataFrame.fillna() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html" \t "_blank) to replace the missing values in the dissatisfied column with the value that occurs most frequently in this column, either True or False.
* Use the [DataFrame.pivot\_table() method](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.pivot_table.html" \t "_blank) to calculate the percentage of dissatisfied employees in each service\_cat group.
  + Since a True value is considered to be 1, calculating the mean will also calculate the percentage of dissatisfied employees. The default aggregation function is the mean, so you can exclude the aggfunc argument.
* Use the [DataFrame.plot()](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html" \t "_blank) method to plot the results. Set the kind parameter equal to bar to create a bar chart.
  + Make sure to run %matplotlib inline beforehand to show your plots in the notebook.
* Write a markdown paragraph briefly describing your observations.

In this guided project, we experienced that in order to extract any meaningful insights from our data, we had to perform many data cleaning tasks. In order to create one visualization (and not even the final one), we completed the following tasks:

* Explored the data and figured out how to prepare it for analysis
* Corrected some of the missing values
* Dropped any data not needed for our analysis
* Renamed our columns
* Verified the quality of our data
* Created a new institute\_service column
* Cleaned the Contributing Factors columns
* Created a new column indicating if an employee resigned because they were dissatisfied in some way
* Combined the data
* Cleaned the institute\_service column
* Handled the missing values in the dissatisfied column
* Aggregated the data

Our work here is far from done! We recommend that you continue with the following steps:

* Decide how to handle the rest of the missing values. Then, aggregate the data according to the service\_cat column again. How many people in each career stage resigned due to some kind of dissatisfaction?
* Clean the age column. How many people in each age group resgined due to some kind of dissatisfaction?
* Instead of analyzing the survey results together, analyze each survey separately. Did more employees in the DETE survey or TAFE survey end their employment because they were dissatisfied in some way?
* Format your project using Dataquest's [project style guide](https://www.dataquest.io/blog/data-science-project-style-guide/).

Curious to see what other students have done on this project? [Head over to our Community to check them out](https://community.dataquest.io/tags/c/social/share/49/348). While you are there, please remember to show some love and give your own feedback!

And of course, we welcome you to share your own project and show off your hard work. Head over to our Community to [share your finished Guided Project](https://community.dataquest.io/tags/c/social/share/49/348)!

Guided projects can be used to build a portfolio to showcase to potential employers, so we really encourage you to keep working on this. You're welcome to keep working on the project here, but we recommend downloading it to your computer using the download icon above and working on it there.

Remember - if you get stuck, don't be afraid to review previous missions or use Google and StackOverflow to search for answers. This isn't easy and will take time and practice.

Congratulations on completing the course! We hope you learned a lot.