

データサイエンス入門とデータサイエンス演習の授業評価分析

- 評価期間: 2022年度後期～2023年度後期
- 分析日: 2024年4月16日
- 分析者: 松田 節郎 (共通教育センター)

1. はじめに

本レポートは、熊本県立大学の全学必修科目であるデータサイエンス入門（1年後期）とデータサイエンス演習（2年前期）について、学期ごとに行われる授業評価アンケートの結果を分析したものである。

受講生には、回答結果が成績評価に影響しないことを知らせた上で、匿名でアンケートに答えてもらっている。また回答方法は、オンラインでのWebアンケート方式となっている。

授業評価アンケートの質問は全部で13項目から成り、「授業の進め方について」の質問が5つ、「受講生の学習状況について」の質問が4つ、「学習理解・満足度について」の質問が4つある。また回答の際には、4～6個ある選択肢のうち、いずれか1つを選ぶリッカート尺度（Likert scale）を採用している。以下では、授業評価において特に重要であると考えられる次の5項目に注目して分析を行う。

- **Q1:** この授業はシラバスに記載された「授業計画」に沿って行われましたか？（選択肢: そう思う, どちらかというと思う, どちらかというと思わない, そう思わない）
- **Q4:** 課題の量は、あなたにとってどうでしたか？（選択肢: 多い, やや多い, 適切, やや少ない, 少ない）
- **Q8:** 授業内容の理解（予習・復習含む）にあてた週の平均時間は何時間くらいですか？（選択肢: 2時間以上, 1～2時間, 30分～1時間, 30分未満, 0分）
- **Q11:** シラバスに記載されているこの授業の「到達目標」は達成されたと思いますか？（選択肢: 達成された, どちらかという達成された, どちらかという達成されていない, 達成されていない）
- **Q13:** この授業の内容を十分に習得できたと実感できましたか？（選択肢: 実感できている, どちらかという実感できている, どちらかという実感できていない, 実感できていない）

回答結果は、累積すると100%になる「積み上げ棒グラフ」として可視化した。またグラフの描画は、筆者が作成した次のPythonコードによって自動化されている。プログラミングに興味のない読者は、読み飛ばしてもらって構わない。

```
In [1]: #----- 必要なライブラリやモジュールの読み込みと設定
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns; sns.set()
import japanize_matplotlib # matplotlibの日本語表示対応
pd.options.display.float_format = '{:.2f}'.format # 浮動小数点数の表示を小数点以下第2位ま
```

```
In [2]: #----- 授業評価アンケート・データの読み込み
df = pd.read_csv('./2022-2023_DS_subjects_evaluation.csv')
print('Dimension of DataFrame:', df.shape)
```

```
#----- 選択肢の取得
multi_choices = df.iloc[0, 7:].to_list()
multi_choices = ['Term', 'Subject'] + multi_choices

#----- 不要な列の削除
df2 = df.drop(['科目NO', '教員名', '履修人数', '回答者数', '回答率'], axis=1)
df2.dropna(inplace=True) # 選択肢行 (欠損値を含む) を削除
df2.columns = multi_choices
df2.reset_index(inplace=True, drop=True)
```

Dimension of DataFrame: (12, 69)

```
In [3]: #----- 教科名 (Subject) と開講学期 (Term) でグループ化
# Subject列とTerm列は、データフレームのMultiIndexとなる
df3 = df2.groupby(['Subject', 'Term']).max()
df3 = df3.astype('int64') # 回答データ (人) を明示的に整数化
print('Dimension of DataFrame:', df3.shape)
```

Dimension of DataFrame: (11, 62)

```
In [4]: #----- 各列の回答データを質問項目別に分ける
# 質問項目: q1~q13 (13項目)
```

```
#----- 授業の進め方について
df_q1 = df3.iloc[:, 0:4]
df_q2 = df3.iloc[:, 4:9]
df_q3 = df3.iloc[:, 9:14]
df_q4 = df3.iloc[:, 14:19]
df_q5 = df3.iloc[:, 19:24]

#----- あなたの学習状況について
df_q6 = df3.iloc[:, 24:30]
df_q7 = df3.iloc[:, 30:36]
df_q8 = df3.iloc[:, 36:41]
df_q9 = df3.iloc[:, 41:46]

#----- 学習理解・満足度について
df_q10 = df3.iloc[:, 46:50]
df_q11 = df3.iloc[:, 50:54]
df_q12 = df3.iloc[:, 54:58]
df_q13 = df3.iloc[:, 58:62]

# pd.DataFrameの行数を取得しておく
nrows = len(df_q1)
print('Number of rows:', nrows)
```

Number of rows: 11

```
In [5]: # 5つの質問に対して、回答データの単位を人数 (人) から割合 (%) に変換
#----- Q1: シラバスに沿って行われたか
row_sums = df_q1.sum(axis=1).values # 各行の合計人数

# 行ごとに割合に変換
for i in range(nrows):
    df_q1.iloc[i, :] = df_q1.iloc[i, :] / row_sums[i] * 100

#----- Q4: 課題の量
row_sums = df_q4.sum(axis=1).values # 各行の合計人数

# 行ごとに割合に変換
for i in range(nrows):
    df_q4.iloc[i, :] = df_q4.iloc[i, :] / row_sums[i] * 100

#----- Q8: 授業内容の理解 (予習・復習にあてた週の平均時間)
row_sums = df_q8.sum(axis=1).values # 各行の合計人数

# 行ごとに割合に変換
for i in range(nrows):
    df_q8.iloc[i, :] = df_q8.iloc[i, :] / row_sums[i] * 100

#----- Q11: 到達目標の達成
row_sums = df_q11.sum(axis=1).values # 各行の合計人数
```

```
# 行ごとに割合に変換
for i in range(nrows):
    df_q11.iloc[i, :] = df_q11.iloc[i, :] / row_sums[i] * 100

#----- Q13: 授業内容の習得
row_sums = df_q13.sum(axis=1).values # 各行の合計人数

# 行ごとに割合に変換
for i in range(nrows):
    df_q13.iloc[i, :] = df_q13.iloc[i, :] / row_sums[i] * 100
```

上記のようなデータの前処理 (preprocessing) を行った上で、授業評価アンケートのQ1, Q4, Q8, Q11, Q13について、回答結果の積み上げ棒グラフを描くと次のようになる。

```
In [6]: #----- 5つの質問に対する回答結果を積み上げ棒グラフで可視化
df_q1.plot(kind='barh', rot=0, title='図1. Q1: シラバスに沿って行われたか',
           figsize=(8, 6), stacked=True)
df_q4.plot(kind='barh', rot=0, title='図2. Q4: 課題の量', figsize=(8, 6), stacked=True)
df_q8.plot(kind='barh', rot=0, title='図3. Q8: 予習・復習にあてた週の平均時間',
           figsize=(8, 6), stacked=True)
df_q11.plot(kind='barh', rot=0, title='図4. Q11: 到達目標の達成', figsize=(8, 6), stacked=True)
df_q13.plot(kind='barh', rot=0, title='図5. Q13: 授業内容の習得', figsize=(8, 6), stacked=True)
```

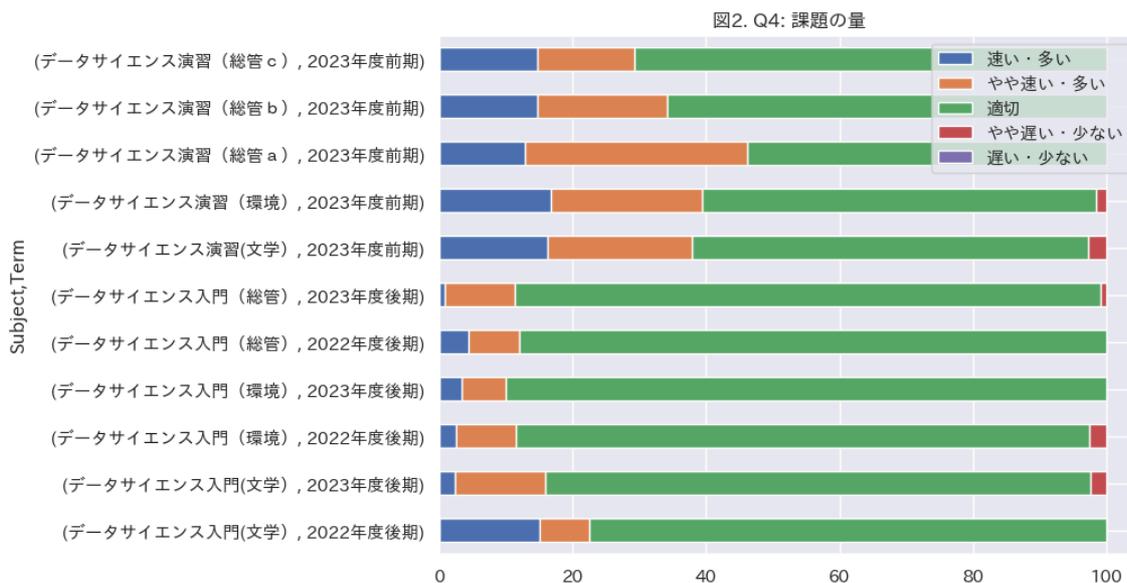
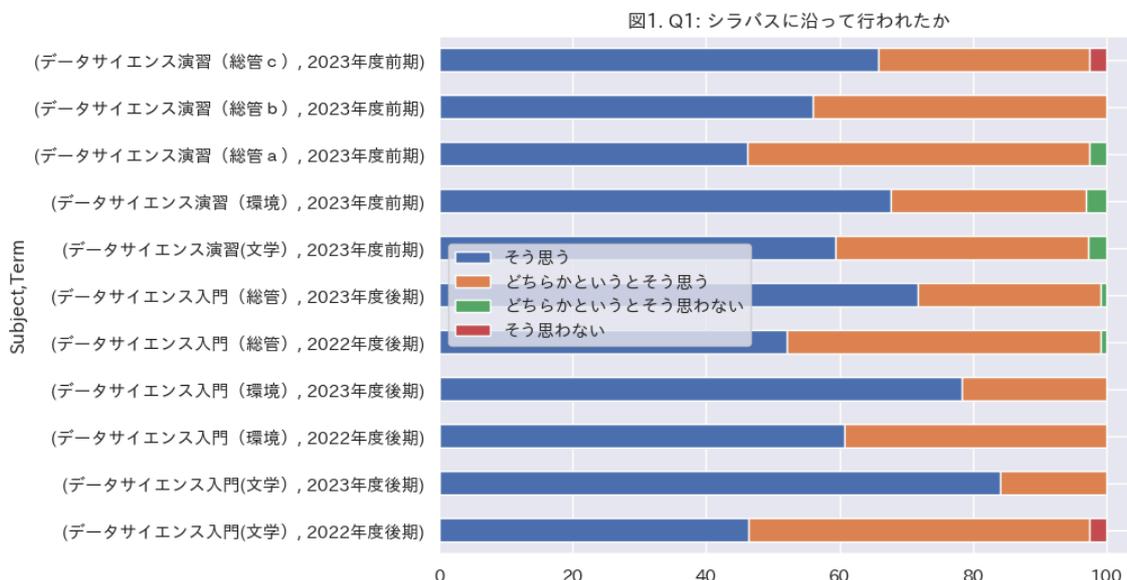


図3. Q8: 予習・復習にあてた週の平均時間

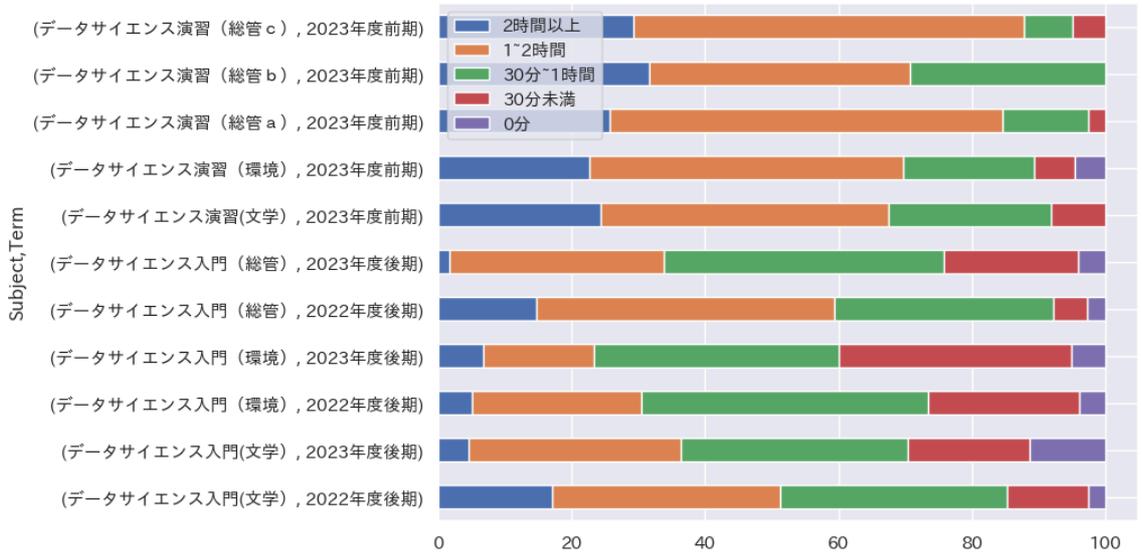


図4. Q11: 到達目標の達成

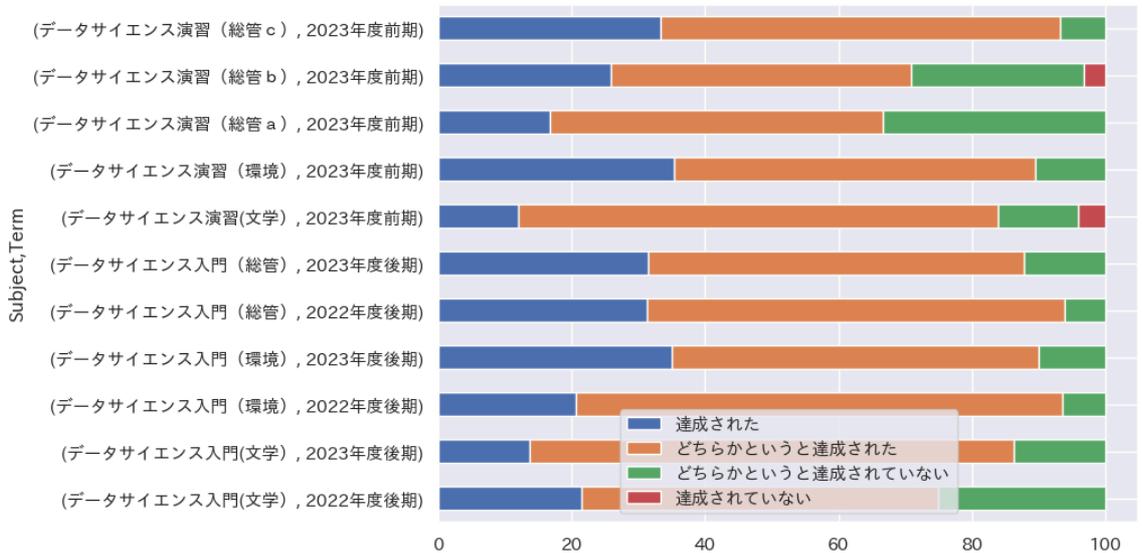
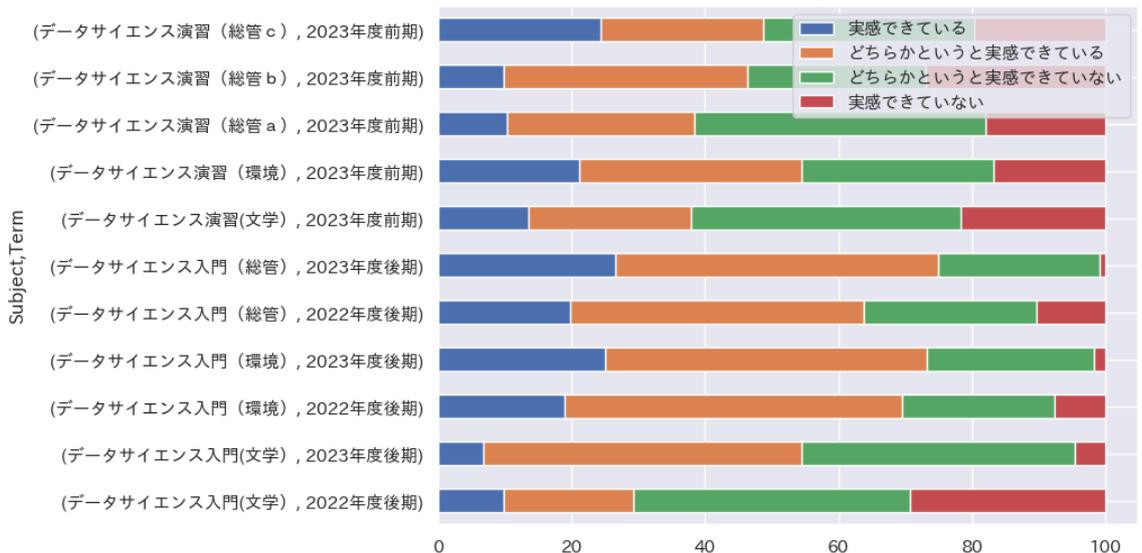


図5. Q13: 授業内容の習得



2. 結果と考察

アンケートの回答率は学部（文学部・環境共生学部・総合管理学部の3学部）によって差があり、約40~60%である。データサイエンス入門は開講2年目なので2学期分のデータがあるが、データサイエンス演習は今年度開講したばかりなので、1学期分のデータしかない点に留意されたい。

まず、授業がシラバスに沿って行われたかを問う図1に注目すると、データサイエンス入門では「そう思う」の割合が前年度より大幅に増加しており、その傾向は特に文学部において顕著である。これは、数式による説明が多かった2022年度に比べて、数式を極力減らしてイメージを伝えるよう修正した2023年度の方が、文学部の学生にとって、授業内容を理解しやすかったことが影響したものと考えられる。データサイエンス演習については、比較可能なデータがないが、「そう思う」と「どちらかというと思う」を合わせた割合が90%を超えているので、大きな問題はないと思われる。

次に、課題の量について問う図2に注目する。データサイエンス入門では、2022年度・2023年度ともに、ほぼ80%の受講生が「適切」と答えている。一方、データサイエンス演習では、約30～40%の受講生が「多い」「やや多い」と回答しており、実データを用いた分析にグループで取り組ませた最終課題がやや負担に感じられたのかもしれない。

受講生の週あたりの平均学習時間を問う図3を見ると、座学のデータサイエンス入門に比べて、実践演習が中心のデータサイエンス演習で「2時間以上」または「1～2時間」と回答した受講生が多いことがわかる。またデータサイエンス入門では、2023年度の難易度が2022年度よりも下がった影響だろうか、「30分未満」または「0分」と回答した割合が増加している。この傾向は特に環境共生学部において顕著であり、本学唯一の理系学部である環境共生学部の学生にとっては、やや物足りない授業内容となった可能性がある。3学部共通の教材を使用しているため、やむを得ない事情もあるが、毎週行っている小テストの問題数を増やす等の工夫が必要だろう。

到達目標の達成度を問う図4を見ると、データサイエンス入門では、2022年度と2023年度で、回答結果にそれほど大きな差はない。しかしながら、文学部の学生については、「達成された」と「どちらかという達成された」を合わせた割合がやや増加し、ここでも授業内容の修正の効果が認められる。データサイエンス演習では、総合管理学部のaクラス（総管a）とbクラス（総管b）で、「どちらかという達成されていない」と回答した受講生がやや多い。これらは非常勤講師に担当いただいたクラスであったため、質問対応が可能な時間帯が制限され、受講生に対するサポートが物理的に困難であった側面があるかもしれない。

最後に、授業内容の習得についての実感を問う図5に注目する。データサイエンス入門では、2022年度から2023年度にかけて、3学部とも「実感できている」と「どちらかという実感できている」を合わせた割合が増加し、「実感できていない」の割合が大幅に減少していることから、上述のように授業内容を修正した効果が認められる。一方、データサイエンス演習では、「どちらかという実感できていない」と「実感できていない」を合わせた割合が、いずれの学部でも半数近くを占めていることがわかる。データサイエンス演習は、深層学習（deep learning）を利用したAIの実装まで踏み込んだ内容となっており、学部2年生には難易度が高かったことが窺える。これを受けて、2024年度のデータサイエンス演習では、深層学習を扱う代わりにテキストマイニングを導入し、文学部の学生の興味・関心を惹きつける内容に修正している。その効果については、来年度のレポートで報告する予定である。

3. 結論

データサイエンス入門とデータサイエンス演習の授業評価アンケートの回答結果を「100%積み上げ棒グラフ」で可視化し、グラフから読み取れることを分析した。その結果をまとめると、次の通りである。

- データサイエンス入門は、授業内容を修正したことによって明らかに改善されている
- データサイエンス入門では、難易度の低下によって物足りなさを感じている受講生もいると考えられるため、課題の量を少し増やす等の改善が必要である
- データサイエンス演習については、高い難易度が問題となっていたため、学生の実態に即した易化が求められる