# 応用数学総合課題

4-J-22 佐藤 優太

## 1. 推定と検定による標本データ解析

### 1) 分析目的

ここでは、性別・年齢別の身長と体重の標本データから母平均の区間推定を行う。

このデータを選んだ理由は、他のデータと比べて分析が行いやすく、授業内容の理解を促進できると考えたからである。また、年齢を重ねるにつれて身長と体重がどのように変わっていくのかを確認する。

### 2)データの特性

推定する母集団は全国民である。この標本データには何も注釈や説明がなかったため、無作為抽出された集団と断言することはできないが、スポーツ庁が担当した調査のためある程度の信頼はおけると考えた。

元のデータは以下の表1のようになっている。

表1: 加工前データの一部

| | 身長 | | | | | | 体重 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | (cm) | | | | | | (kg) | | | | | |
| 年齢 | 男子 | | | 女子 | | | 男子 | | | 女子 | | |
| | 標本数 | 平均値 | 標準偏差 | 標本数 | 平均値 | 標準偏差 | 標本数 | 平均値 | 標準偏差 | 標本数 | 平均値 | 標準偏差 |
| 6 | 1111 | 116.62 | 4.88 | 1107 | 115.64 | 4.66 | 1089 | 21.26 | 2.85 | 1084 | 20.79 | 2.82 |
| 7 | 1109 | 122.44 | 5.04 | 1111 | 121.63 | 5.10 | 1087 | 23.81 | 3.35 | 1081 | 23.21 | 3.14 |
| 8 | 1125 | 128.33 | 5.18 | 1115 | 127.38 | 5.20 | 1088 | 26.80 | 4.02 | 1093 | 26.32 | 4.17 |
| 9 | 1112 | 133.50 | 5.46 | 1108 | 133.59 | 6.20 | 1083 | 29.98 | 4.94 | 1091 | 29.81 | 5.01 |
| 10 | 1116 | 138.80 | 5.88 | 1117 | 140.25 | 6.92 | 1087 | 33.43 | 5.89 | 1096 | 33.92 | 6.33 |
| 11 | 1113 | 145.53 | 7.03 | 1118 | 147.13 | 6.48 | 1093 | 38.01 | 7.17 | 1106 | 38.87 | 6.86 |
| 12 | 1377 | 152.81 | 8.07 | 1379 | 151.98 | 5.94 | 1357 | 43.44 | 8.27 | 1360 | 43.35 | 7.12 |
| 13 | 1370 | 160.75 | 7.44 | 1378 | 155.06 | 5.41 | 1342 | 48.78 | 8.37 | 1345 | 46.38 | 6.52 |
| 14 | 1377 | 165.96 | 6.28 | 1386 | 156.36 | 5.24 | 1359 | 54.04 | 8.02 | 1366 | 48.81 | 6.48 |
| 15 | 1411 | 168.37 | 5.75 | 1413 | 156.76 | 5.36 | 1377 | 57.40 | 8.83 | 1381 | 50.82 | 6.53 |
| 16 | 1428 | 169.59 | 5.70 | 1419 | 157.16 | 5.17 | 1387 | 59.45 | 8.45 | 1387 | 51.81 | 6.66 |
| 17 | 1427 | 170.46 | 5.82 | 1431 | 157.13 | 5.34 | 1388 | 61.58 | 8.84 | 1402 | 51.53 | 6.53 |
| 18 | 911 | 171.10 | 5.66 | 999 | 157.76 | 5.40 | 888 | 62.16 | 8.45 | 955 | 51.57 | 6.43 |
| 19 | 737 | 171.59 | 5.65 | 672 | 157.86 | 5.12 | 718 | 62.86 | 8.15 | 655 | 51.73 | 6.16 |
| 20–24 | 1269 | 171.50 | 5.55 | 1025 | 158.49 | 5.24 | 1239 | 65.74 | 8.87 | 934 | 50.87 | 5.95 |
| 25–29 | 1334 | 172.05 | 5.61 | 968 | 158.94 | 5.22 | 1295 | 67.21 | 9.28 | 864 | 50.73 | 5.76 |
| 30–34 | 1298 | 172.12 | 5.67 | 1061 | 158.70 | 5.27 | 1271 | 68.69 | 9.53 | 934 | 51.39 | 6.13 |
| 35–39 | 1451 | 172.31 | 5.57 | 1384 | 158.96 | 5.24 | 1435 | 68.80 | 9.33 | 1212 | 51.79 | 6.14 |

### 3)分析と分析結果

まず、ファイルを読み込み、データを身長と体重に分けてデータフレームに格納し、ヘッダーを設定した。ソースコードとデータフレームは以下の図1に示す。

```
col_names = ["age", "M_sample", "M_mean", "M_sd", "F_sample", "F_mean", "F_sd"]
height = pd.read_excel("data.xlsx", sheet_name=0, skiprows=4, usecols=range(7), names=col_names)
weight = pd.read_excel("data.xlsx", sheet_name=0, skiprows=4, usecols=[0, 7, 8, 9, 10, 11, 12], names=col_names)
```

```
height.head()
```

| | age | M_sample | M_mean | M_sd | F_sample | F_mean | F_sd |
|---|---|---|---|---|---|---|---|
| 0 | 6 | 1111 | 116.62 | 4.88 | 1107 | 115.64 | 4.66 |
| 1 | 7 | 1109 | 122.44 | 5.04 | 1111 | 121.63 | 5.10 |
| 2 | 8 | 1125 | 128.33 | 5.18 | 1115 | 127.38 | 5.20 |
| 3 | 9 | 1112 | 133.50 | 5.46 | 1108 | 133.59 | 6.20 |
| 4 | 10 | 1116 | 138.80 | 5.88 | 1117 | 140.25 | 6.92 |

```
weight.head()
```

| | age | M_sample | M_mean | M_sd | F_sample | F_mean | F_sd |
|---|---|---|---|---|---|---|---|
| 0 | 6 | 1089 | 21.26 | 2.85 | 1084 | 20.79 | 2.82 |
| 1 | 7 | 1087 | 23.81 | 3.35 | 1081 | 23.21 | 3.14 |
| 2 | 8 | 1088 | 26.80 | 4.02 | 1093 | 26.32 | 4.17 |
| 3 | 9 | 1083 | 29.98 | 4.94 | 1091 | 29.81 | 5.01 |
| 4 | 10 | 1087 | 33.43 | 5.89 | 1096 | 33.92 | 6.33 |

図1: データフレームの作成

ここで各カラム名について説明する。ageは年齢である。M、Fはそれぞれ男性、女性を表しており、sampleは標本データ数、meanは標本平均(単位: cm)、sdは標本標準偏差を表している。

次に、身長と体重それぞれのデータフレームにおいて、男性と女性それぞれの不偏分散を求め、新しくudというカラムを作成しそこに格納した。

以上のデータを用いて信頼限界を求めた。信頼下界はpm_under、信頼上界はpm_overというカラムに格納した。ここまでのソースコードとデータフレームを以下の図2と図3に示す。

```
# 不偏分散を求める
height.insert(4, "M_ud", np.sqrt((height["M_sample"] / (height["M_sample"] - 1)) * height["M_sd"] ** 2))
height.insert(8, "F_ud", np.sqrt((height["F_sample"] / (height["F_sample"] - 1)) * height["F_sd"] ** 2))

#信頼限界を求める
height.insert(5, "M_pm_under", height["M_mean"] - 2.262 * (height["M_ud"] / np.sqrt(height["M_sample"])))
height.insert(6, "M_pm_over", height["M_mean"] + 2.262 * (height["M_ud"] / np.sqrt(height["M_sample"])))
height.insert(11, "F_pm_under", height["F_mean"] - 2.262 * (height["F_ud"] / np.sqrt(height["F_sample"])))
height.insert(12, "F_pm_over", height["F_mean"] + 2.262 * (height["F_ud"] / np.sqrt(height["F_sample"])))
```

```
height.head()
```

| | age | M_sample | M_mean | M_sd | M_ud | M_pm_under | M_pm_over | F_sample | F_mean | F_sd | F_ud | F_pm_under | F_pm_over |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 1111 | 116.62 | 4.88 | 4.882198 | 116.288677 | 116.951323 | 1107 | 115.64 | 4.66 | 4.662106 | 115.323043 | 115.956957 |
| 1 | 7 | 1109 | 122.44 | 5.04 | 5.042274 | 122.097506 | 122.782494 | 1111 | 121.63 | 5.10 | 5.102297 | 121.283741 | 121.976259 |
| 2 | 8 | 1125 | 128.33 | 5.18 | 5.182304 | 127.980506 | 128.679494 | 1115 | 127.38 | 5.20 | 5.202333 | 127.027586 | 127.732414 |
| 3 | 9 | 1112 | 133.50 | 5.46 | 5.462457 | 133.129466 | 133.870534 | 1108 | 133.59 | 6.20 | 6.202800 | 133.168487 | 134.011513 |
| 4 | 10 | 1116 | 138.80 | 5.88 | 5.882636 | 138.401680 | 139.198320 | 1117 | 140.25 | 6.92 | 6.923100 | 139.781439 | 140.718561 |

図2: 身長のデータフレーム

```
# 不偏分散を求める
weight.insert(4, "M_ud", (weight["M_sample"] / (weight["M_sample"] - 1)) * weight["M_sd"])
weight.insert(8, "F_ud", (weight["F_sample"] / (weight["F_sample"] - 1)) * weight["F_sd"])

#信頼限界を求める
weight.insert(5, "M_pm_under", weight["M_mean"] - 2.262 * (weight["M_ud"] / np.sqrt(weight["M_sample"])))
weight.insert(6, "M_pm_over", weight["M_mean"] + 2.262 * (weight["M_ud"] / np.sqrt(weight["M_sample"])))
weight.insert(11, "F_pm_under", weight["F_mean"] - 2.262 * (weight["F_ud"] / np.sqrt(weight["F_sample"])))
weight.insert(12, "F_pm_over", weight["F_mean"] + 2.262 * (weight["F_ud"] / np.sqrt(weight["F_sample"])))
```

```
weight.head()
```

| | age | M_sample | M_mean | M_sd | M_ud | M_pm_under | M_pm_over | F_sample | F_mean | F_sd | F_ud | F_pm_under | F_pm_over |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 1089 | 21.26 | 2.85 | 2.852619 | 21.064466 | 21.455534 | 1084 | 20.79 | 2.82 | 2.822604 | 20.596078 | 20.983922 |
| 1 | 7 | 1087 | 23.81 | 3.35 | 3.353085 | 23.579950 | 24.040050 | 1081 | 23.21 | 3.14 | 3.142907 | 22.993772 | 23.426228 |
| 2 | 8 | 1088 | 26.80 | 4.02 | 4.023698 | 26.524067 | 27.075933 | 1093 | 26.32 | 4.17 | 4.173819 | 26.034428 | 26.605572 |
| 3 | 9 | 1083 | 29.98 | 4.94 | 4.944566 | 29.640135 | 30.319865 | 1091 | 29.81 | 5.01 | 5.014596 | 29.466587 | 30.153413 |
| 4 | 10 | 1087 | 33.43 | 5.89 | 5.895424 | 33.025524 | 33.834476 | 1096 | 33.92 | 6.33 | 6.335781 | 33.487100 | 34.352900 |

図3: 体重のデータフレーム

最後に、身長と体重それぞれについて、男性と女性それぞれの信頼限界をプロットした。ソースコードとプロット結果を以下の図4に示す。

```
height["age"].iloc[14:] = height["age"].iloc[14:].str[:2]
plt.plot(height["age"], height["M_pm_under"], label="M_pm_under")
plt.plot(height["age"], height["M_pm_over"], label="M_pm_over")
plt.plot(height["age"], height["F_pm_under"], label="F_pm_under")
plt.plot(height["age"], height["F_pm_over"], label="F_pm_over")
plt.xlabel("age")
plt.ylabel("height")
plt.legend()
plt.grid()
plt.show()
```

```
weight["age"].iloc[14:] = weight["age"].iloc[14:].str[:2]
plt.plot(weight["age"], weight["M_pm_under"], label="M_pm_under")
plt.plot(weight["age"], weight["M_pm_over"], label="M_pm_over")
plt.plot(weight["age"], weight["F_pm_under"], label="F_pm_under")
plt.plot(weight["age"], weight["F_pm_over"], label="F_pm_over")
plt.xlabel("age")
plt.ylabel("weight")
plt.legend()
plt.grid()
plt.show()
```

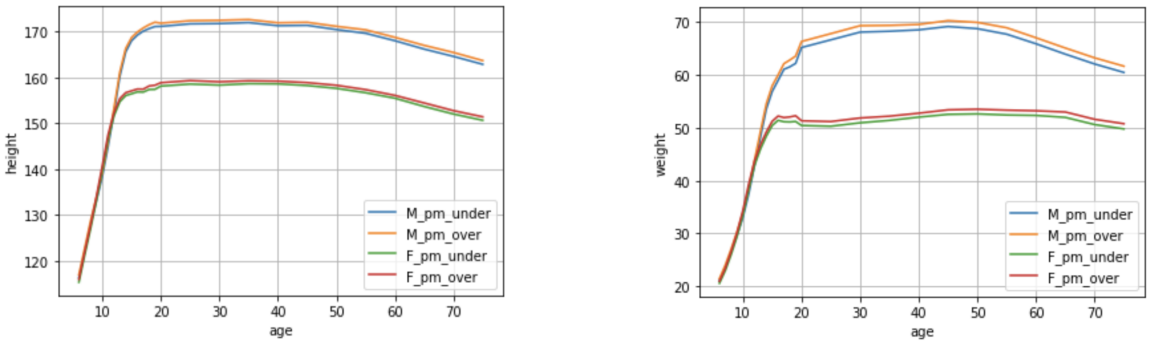


図4: 信頼限界のプロット(左: 身長, 右: 体重)

## 4)分析結果の考察

母平均の信頼区間の幅がとても狭かったが、これは標本データ数が1000を超えているために非常に精度が高い結果が得られたということだと解釈できる。

身長は、男性も女性もおよそ15歳までに急激に増加し、15〜20歳の間に緩やかに増加、その後は緩やかに減少していくという結果になった。

体重について、男性はおよそ20歳までに急激に増加し、20〜45歳の間に緩やかに増加、その後は減少していくという結果になった。女性はおよそ15歳までに急激に増加し、15〜25歳の間に一度緩やかに減少するが、25〜45歳頃まで緩やかに増加し、その後、緩やかに減少するという結果になった。

男性も女性も身長は20歳頃からだんだん減少し始めるが、体重は緩やかに増加しているため、これがいわゆる中年太りだと考えられる。また、女性の体重が15〜25歳の間で一度減少しているのは、この時期にダイエットを始め、痩せようとする女性が多いからだと考えられる。

## 5)データ出典

スポーツ庁調査統計企画室. 体力・運動能力調査 / 平成３０年度 年齢別体格測定の結果 身長、体重. e-Stat. 閲覧日: 2020-08-06.
https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei=00402102&bunya_l=12&tstat=000001088875&cycle=0&tclass1=000001133904&stat_infid=000031872003

# 2. 実データに対するフーリエ変換

## 1)分析目的

　ここでは、江刺の2015年1月から2019年12月までの気温データをフーリエ変換する。
　このデータを選んだ理由は、他の時系列データよりも分析が行いやすく、授業内容の理解が促進できると考えたからである。また、四季による気温の移り変わりにある程度の周期性があることを確認する。

## 2)データの特性

　元のデータには江刺の2015年1月から2019年12月までの1ヶ月ごとの気温データが格納されている。各データは小数点第一位まで表示されている。また、気温データとともに品質情報と均質番号が付与されているが、全てのデータに対して、品質番号は8、均質番号は1である。これは、データに欠損がないことと、観測環境に変化がないことを表してる。

　元のデータは以下の表2のようになっている。

表2: 加工前データの一部

| ダウンロードした時刻：2020/08/09 11:20:16 | | | |
|---|---|---|---|
| | 江刺 | 江刺 | 江刺 |
| 年月 | 平均気温(℃) | 平均気温(℃) | 平均気温(℃) |
| | | 品質情報 | 均質番号 |
| Jan-15 | 0.1 | 8 | 1 |
| Feb-15 | 1.3 | 8 | 1 |
| Mar-15 | 5.3 | 8 | 1 |
| Apr-15 | 10.9 | 8 | 1 |
| May-15 | 17.3 | 8 | 1 |
| Jun-15 | 19.9 | 8 | 1 |
| Jul-15 | 24.8 | 8 | 1 |
| Aug-15 | 24.1 | 8 | 1 |
| Sep-15 | 19.5 | 8 | 1 |
| Oct-15 | 12.7 | 8 | 1 |
| Nov-15 | 8.2 | 8 | 1 |
| Dec-15 | 2.8 | 8 | 1 |
| Jan-16 | 0.1 | 8 | 1 |
| Feb-16 | 0.8 | 8 | 1 |
| Mar-16 | 4.8 | 8 | 1 |
| Apr-16 | 10.4 | 8 | 1 |
| May-16 | 16.7 | 8 | 1 |
| Jun-16 | 19.6 | 8 | 1 |
| Jul-16 | 22.8 | 8 | 1 |
| Aug-16 | 25.2 | 8 | 1 |

## 3)分析と分析結果

　まず、ファイルを読み込み、年月と気温の列のみをデータフレームに格納し、ヘッダーを設定した。ソースコードとデータフレームは以下の図5に示す。

```
temp = pd.read_csv("temperature.csv", skiprows=5, usecols=range(2), names=["months", "temperature"])
temp.head()
```

| | months | temperature |
|---|---|---|
| 0 | 2015/1 | 0.1 |
| 1 | 2015/2 | 1.3 |
| 2 | 2015/3 | 5.3 |
| 3 | 2015/4 | 10.9 |
| 4 | 2015/5 | 17.3 |

図5: データフレームの作成

ここでカラム名について説明する。monthsは年月、temperatureは気温(単位: °C)を表している。

次に、気温のデータを折れ線グラフでプロットした。ソースコードとプロット結果は以下の図6に示す。

```
plt.figure(figsize=(10, 4))
plt.xlabel('time(month)', fontsize=14)
plt.ylabel('temperature(℃)', fontsize=14)
plt.xticks(np.arange(0, 61, 3))
plt.grid()
plt.plot(temp["temperature"], marker='o')
```

```
[<matplotlib.lines.Line2D at 0x7fd2edbff650>]
```

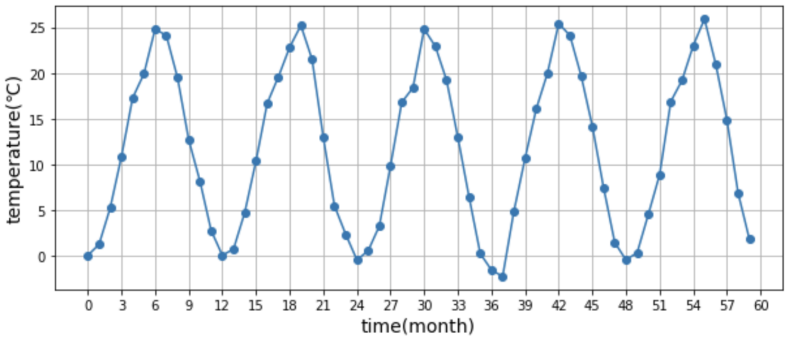


図6: 気温データのプロット

次に、データフレームに格納されている気温データに対して高速フーリエ変換(FFT)を適用した。また、FFT結果は複素数となるため、絶対値に変換した。その後、周期を確認するためにFFT結果をグラフに表示させたが、その際、振幅は元のデータに揃え、また、周波数軸のデータを作成した。ソースコードとプロット結果は以下の図7に示す。

```
# 高速フーリエ変換(FFT)
F = np.fft.fft(temp["temperature"])
# FFT結果を絶対値に変換
F_abs = np.abs(F)
# 振幅を元のデータに揃える
F_abs_amp = F_abs / len(temp.index) * 2
# 周波数軸のデータ作成
fq = np.linspace(0, 1, len(temp.index))

#グラフ表示
plt.xlabel('freqency(Hz)', fontsize=14)
plt.ylabel('amplitude', fontsize=14)
plt.xticks(np.arange(0, 0.6, 0.05))
plt.grid()
plt.plot(fq[:int(len(temp.index)/2)+1], F_abs_amp[:int(len(temp.index)/2)+1]) # ナイキスト定数まで表示
```

```
[<matplotlib.lines.Line2D at 0x7fd72412d190>]
```

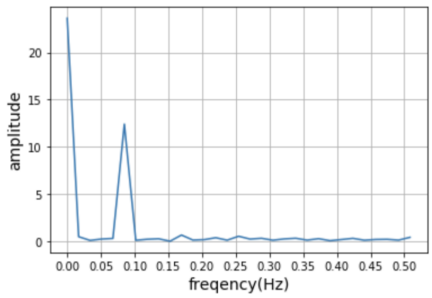


図7: FFT結果のプロット

最後に、FFT結果に対して逆高速フーリエ変換(IFFT)を適用し、その結果と元のデータを重ねてグラフ表示させ、FFT結果が正しいかどうかを確認した。元のデータの値とIFFT結果が一致しているので、FFT結果が正しいことがわかる。ソースコードとプロット結果は以下の図8に示す。

```
#逆高速フーリエ変換(IFFT)
F_ifft = np.fft.ifft(F)
#実数部の取り出し
F_ifft_real = F_ifft.real

#グラフ表示
plt.figure(figsize=(10, 4))
plt.xlabel('time(month)', fontsize=14)
plt.ylabel('temperature(℃)', fontsize=14)
plt.xticks(np.arange(0, 61, 3))
plt.grid()
plt.plot(F_ifft_real, marker='o', label="IFFT")#IFFT結果
plt.plot(temp["temperature"], marker='o', linestyle="--", label="original") #元データ
plt.legend(loc='best')
```

```
<matplotlib.legend.Legend at 0x7fd7247e2f50>
```

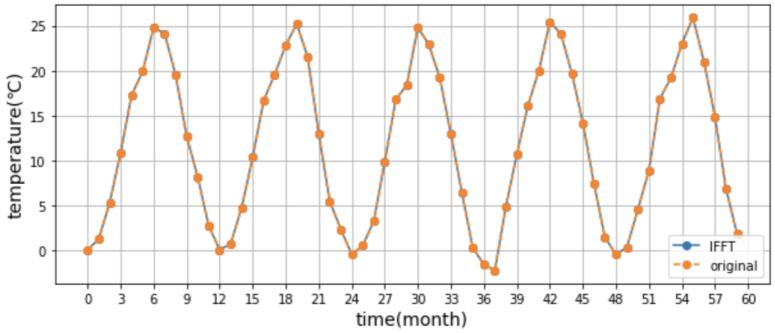


図8: IFFT結果のプロット

## 4)分析結果の考察

図7: FFT結果のプロットより、周波数がおよそ0Hzと0.08Hzのところでピークが出ていることがわかる。このように、プロット結果が連続値ではなく離散値となったので、今回扱った気温データには周期性があると考えられる。

## 5)データ出典

江刺 月平均気温 2015年1月から2019年12月までの月別値. 気象庁. 閲覧日: 2020-08-09.
https://www.data.jma.go.jp/gmd/risk/obsdl/index.php