Brief Analysis on Student Performance Data Set

November, 2020

目次

1 データ情報

【データ名】
Student Performance Data Set

【取得元】
カリフォルニア大学アーバイン校／機械学習リポジトリ [1]

【原著論文】
USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE [2]
(ポルトガルの高校における数学とポルトガル語の学年末成績の予測)

1.1 変数一覧

No.	変数名	内容	詳細
1	school	学校名	binary ‘GP’ - Gabriel Pereira or ‘MS’ - Mousinho da Silveira
2	sex	性別	binary: ‘F’ - female or ‘M’ - male
3	age	年齢	numeric: from 15 to 22
4	address	居住地域	binary: ‘U’ - urban or ‘R’ - rural
5	famsize	世帯人数	binary: ‘LE3’ - less or equal to 3 or ‘GT3’ - greater than 3
6	Pstatus	両親との同居有無	binary: ‘T’ - living together or ‘A’ - apart
7	Medu	母親の学歴	numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education
8	Fedu	父親の学歴	numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education
9	Mjob	母親の職業	nominal: ‘teacher’, ‘health’ care related, civil ‘services’ (e.g. administrative or police), ‘at_home’ or ‘other’
10	Fjob	父親の職業	nominal: ‘teacher’, ‘health’ care related, civil ‘services’ (e.g. administrative or police), ‘at_home’ or ‘other’
11	reason	現在籍校への進学理由	nominal: close to ‘home’, school ‘reputation’, ‘course’ preference or ‘other’
12	guardian	保護者	nominal: ‘mother’, ‘father’ or ‘other’
13	traveltime	通学時間	numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour
14	studytime	週平均学習時間	numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours
15	failures	落とした単位数	numeric: n if 1<=n<3, else 4
16	schoolsup	追加学習サポート	binary: yes or no
17	famsup	家族の学習サポート	binary: yes or no
18	paid	追加有料講座受講	binary: yes or no
19	activities	課外活動	binary: yes or no
20	nursery	保育園歴	binary: yes or no
21	higher	大学進学希望	binary: yes or no
22	internet	自宅ネット環境	binary: yes or no
23	romantic	交際相手	binary: yes or no
24	famrel	家族との関係	numeric: from 1 - very bad to 5 - excellent
25	freetime	放課後の自由時間	numeric: from 1 - very low to 5 - very high
26	goout	交友頻度	numeric: from 1 - very low to 5 - very high
27	Dalc	平日の飲酒	numeric: from 1 - very low to 5 - very high
28	Walc	週末の飲酒	numeric: from 1 - very low to 5 - very high
29	health	健康状態	numeric: from 1 - very bad to 5 - very good
30	absences	欠席日数	numeric: from 0 to 93
31	G1	一学期の成績	numeric: from 0 to 20
31	G2	二学期の成績	numeric: from 0 to 20
32	G3	三学期の成績	numeric: from 0 to 20 ＊目的変数

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='darkgrid', font_scale = 1.8)
plt.rcParams['figure.dpi'] = 300
%matplotlib inline
```

```
Duplicate key in file PosixPath('/Users/sunggi/opt/anaconda3/lib/python3.7/site-
packages/matplotlib/mpl-data/matplotlibrc'), line 250 ('font.family:
IPAexGothic')
```

[2]:
```
df_m = pd.read_csv('./data/student-mat.csv', sep=';')
df_p = pd.read_csv('./data/student-por.csv', sep=';')
```

1.2 データ取得校所在地

[3]:
```
import folium

# Gabriel Pereira校, Address:  Rua Dr. Domingos Rosado  7005-469 Évora
#  https://aegp.edu.pt/web/pt-pt/gabriel-pereira
Pereira = [38.5745, -7.9025]

# Mousinho da Silveira校, Address: Avenida do Bonfim, 7300-067 Portalegre
# https://aeb.pt/portal/agrupamento/escolas/
Silveira = [39.3025, -7.4333]

# center = [(x+y)/2 +10 for (x, y) in zip(Pereira, Silveira)]
center =[45, 5]

m = folium.Map(location=center, tiles='Stamen Terrain', zoom_start=4.5)
folium.Marker(location=Pereira, popup='<b>Gabriel Pereira</b>').add_to(m)
folium.Marker(location=Silveira, popup='<b>Mousinho da Silveira</b>').add_to(m)
m
```

1.3 データの様子

[4]:
```
df_m.head().T
```

[4]:

	0	1	2	3	4
school	GP	GP	GP	GP	GP
sex	F	F	F	F	F
age	18	17	15	15	16
address	U	U	U	U	U
famsize	GT3	GT3	LE3	GT3	GT3
Pstatus	A	T	T	T	T
Medu	4	1	1	4	3
Fedu	4	1	1	2	3
Mjob	at_home	at_home	at_home	health	other
Fjob	teacher	other	other	services	other
reason	course	course	other	home	home
guardian	mother	father	mother	mother	father
traveltime	2	1	1	1	1
studytime	2	2	2	3	2
failures	0	0	3	0	0
schoolsup	yes	no	yes	no	no
famsup	no	yes	no	yes	yes
paid	no	no	yes	yes	yes
activities	no	no	no	yes	no
nursery	yes	no	yes	yes	yes
higher	yes	yes	yes	yes	yes
internet	no	yes	yes	yes	no
romantic	no	no	no	yes	no
famrel	4	5	4	3	4
freetime	3	3	3	2	3
goout	4	3	2	2	2
Dalc	1	1	2	1	1
Walc	1	1	3	1	2
health	3	3	3	5	5
absences	6	4	10	2	4
G1	5	5	7	15	6
G2	6	5	8	14	10
G3	6	6	10	15	10

[5]:
```
df_p.head().T
```

[5]:

	0	1	2	3	4
school	GP	GP	GP	GP	GP
sex	F	F	F	F	F
age	18	17	15	15	16
address	U	U	U	U	U
famsize	GT3	GT3	LE3	GT3	GT3
Pstatus	A	T	T	T	T
Medu	4	1	1	4	3
Fedu	4	1	1	2	3
Mjob	at_home	at_home	at_home	health	other
Fjob	teacher	other	other	services	other
reason	course	course	other	home	home
guardian	mother	father	mother	mother	father
traveltime	2	1	1	1	1
studytime	2	2	2	3	2
failures	0	0	0	0	0
schoolsup	yes	no	yes	no	no
famsup	no	yes	no	yes	yes
paid	no	no	no	no	no
activities	no	no	no	yes	no
nursery	yes	no	yes	yes	yes
higher	yes	yes	yes	yes	yes
internet	no	yes	yes	yes	no
romantic	no	no	no	yes	no
famrel	4	5	4	3	4
freetime	3	3	3	2	3
goout	4	3	2	2	2
Dalc	1	1	2	1	1
Walc	1	1	3	1	2
health	3	3	3	5	5
absences	4	2	6	0	0
G1	0	9	12	14	11
G2	11	11	13	14	13
G3	11	11	12	14	13

[6]:
```
# df_m.info()
```

[7]:
```
# (df_m.dtypes == df_p.dtypes).sum()
```

[8]:
```
# df_m.isna().sum().sum()
```

[9]:
```
# df_p.isna().sum().sum()
```

2　基本統計量

[10]:
```
cat = []
num = []
for i in range(df_m.shape[1]):
    if df_m.dtypes.values[i] == 'O':
        cat.append(df_m.columns[i])
    else:
        num.append(df_m.columns[i])
print('NOE: cat=', len(cat))
print('NOE: num=', len(num))
```

```
NOE: cat= 17
NOE: num= 16
```

2.1　質的変数

[11]:
```
print('数学')
display(df_m[cat].describe().T)
print('\n', 'ポルトガル語')
display(df_p[cat].describe().T)
```

数学

	count	unique	top	freq
school	395	2	GP	349
sex	395	2	F	208
address	395	2	U	307
famsize	395	2	GT3	281
Pstatus	395	2	T	354
Mjob	395	5	other	141
Fjob	395	5	other	217
reason	395	4	course	145
guardian	395	3	mother	273
schoolsup	395	2	no	344
famsup	395	2	yes	242
paid	395	2	no	214
activities	395	2	yes	201
nursery	395	2	yes	314
higher	395	2	yes	375
internet	395	2	yes	329
romantic	395	2	no	263

ポルトガル語

	count	unique	top	freq
school	649	2	GP	423
sex	649	2	F	383
address	649	2	U	452
famsize	649	2	GT3	457
Pstatus	649	2	T	569
Mjob	649	5	other	258
Fjob	649	5	other	367
reason	649	4	course	285
guardian	649	3	mother	455
schoolsup	649	2	no	581
famsup	649	2	yes	398
paid	649	2	no	610
activities	649	2	no	334
nursery	649	2	yes	521
higher	649	2	yes	580
internet	649	2	yes	498
romantic	649	2	no	410

2.2 量的変数

[12]:
```
print('数学')
display(round(df_m[num].describe(),1).T)
print('\n', 'ポルトガル語')
display(round(df_p[num].describe(),1).T)
```

数学

	count	mean	std	min	25%	50%	75%	max
age	395.0	16.7	1.3	15.0	16.0	17.0	18.0	22.0
Medu	395.0	2.7	1.1	0.0	2.0	3.0	4.0	4.0
Fedu	395.0	2.5	1.1	0.0	2.0	2.0	3.0	4.0
traveltime	395.0	1.4	0.7	1.0	1.0	1.0	2.0	4.0
studytime	395.0	2.0	0.8	1.0	1.0	2.0	2.0	4.0
failures	395.0	0.3	0.7	0.0	0.0	0.0	0.0	3.0
famrel	395.0	3.9	0.9	1.0	4.0	4.0	5.0	5.0
freetime	395.0	3.2	1.0	1.0	3.0	3.0	4.0	5.0
goout	395.0	3.1	1.1	1.0	2.0	3.0	4.0	5.0
Dalc	395.0	1.5	0.9	1.0	1.0	1.0	2.0	5.0
Walc	395.0	2.3	1.3	1.0	1.0	2.0	3.0	5.0
health	395.0	3.6	1.4	1.0	3.0	4.0	5.0	5.0
absences	395.0	5.7	8.0	0.0	0.0	4.0	8.0	75.0
G1	395.0	10.9	3.3	3.0	8.0	11.0	13.0	19.0
G2	395.0	10.7	3.8	0.0	9.0	11.0	13.0	19.0
G3	395.0	10.4	4.6	0.0	8.0	11.0	14.0	20.0

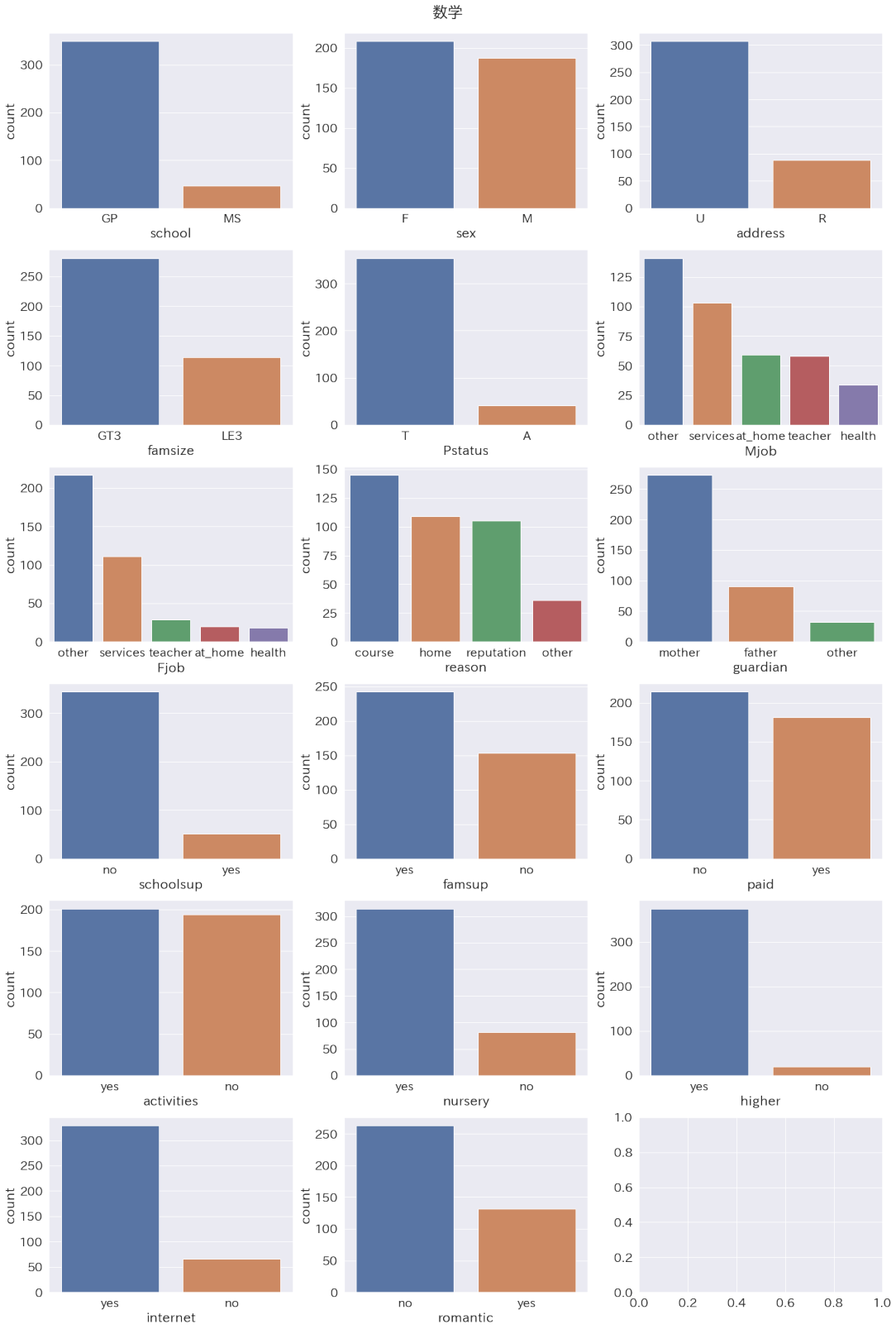
ポルトガル語

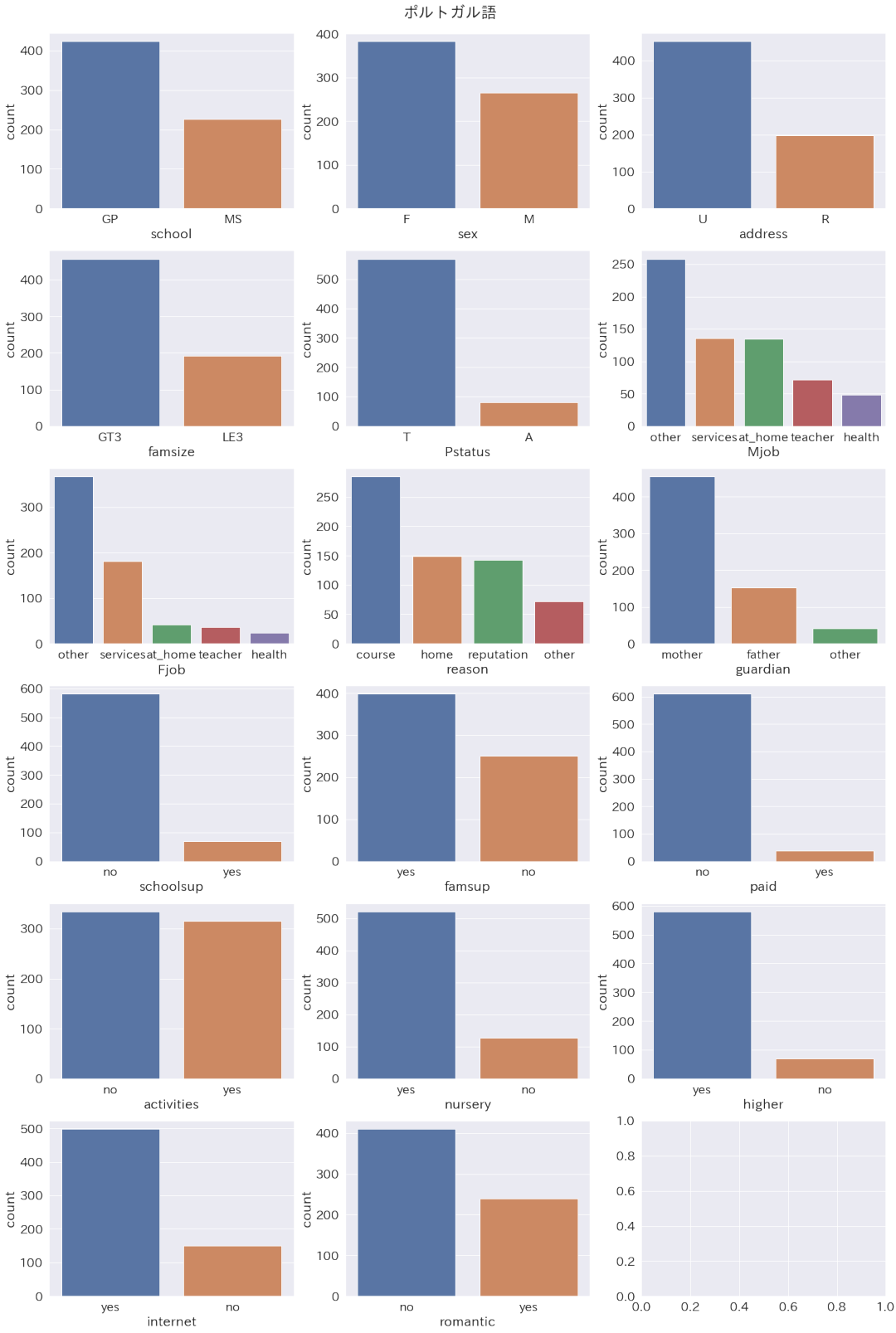
	count	mean	std	min	25%	50%	75%	max
age	649.0	16.7	1.2	15.0	16.0	17.0	18.0	22.0
Medu	649.0	2.5	1.1	0.0	2.0	2.0	4.0	4.0
Fedu	649.0	2.3	1.1	0.0	1.0	2.0	3.0	4.0
traveltime	649.0	1.6	0.7	1.0	1.0	1.0	2.0	4.0
studytime	649.0	1.9	0.8	1.0	1.0	2.0	2.0	4.0
failures	649.0	0.2	0.6	0.0	0.0	0.0	0.0	3.0
famrel	649.0	3.9	1.0	1.0	4.0	4.0	5.0	5.0
freetime	649.0	3.2	1.1	1.0	3.0	3.0	4.0	5.0
goout	649.0	3.2	1.2	1.0	2.0	3.0	4.0	5.0
Dalc	649.0	1.5	0.9	1.0	1.0	1.0	2.0	5.0
Walc	649.0	2.3	1.3	1.0	1.0	2.0	3.0	5.0
health	649.0	3.5	1.4	1.0	2.0	4.0	5.0	5.0
absences	649.0	3.7	4.6	0.0	0.0	2.0	6.0	32.0
G1	649.0	11.4	2.7	0.0	10.0	11.0	13.0	19.0
G2	649.0	11.6	2.9	0.0	10.0	11.0	13.0	19.0
G3	649.0	11.9	3.2	0.0	10.0	12.0	14.0	19.0

3　探索的分析：質的データ

3.1　【作例】度数の表示

[13]:
```
nor = len(cat)//3+1
noc = 3
for df1, sub in zip([df_m, df_p], ['数学', 'ポルトガル語']):
    fig, axs = plt.subplots(nor, noc, figsize=(20, 30))
    for i in range(nor):
        for j in range(noc):
            k = 3*i + j
            if k < len(cat):
                sns.countplot(x=cat[k], data=df1, order=df1[cat[k]].value_counts().index,␣
↪ax=axs[i, j])
    fig.tight_layout()
    fig.suptitle(sub, fontsize=25)
    fig.subplots_adjust(top=0.96)
    plt.show()
```

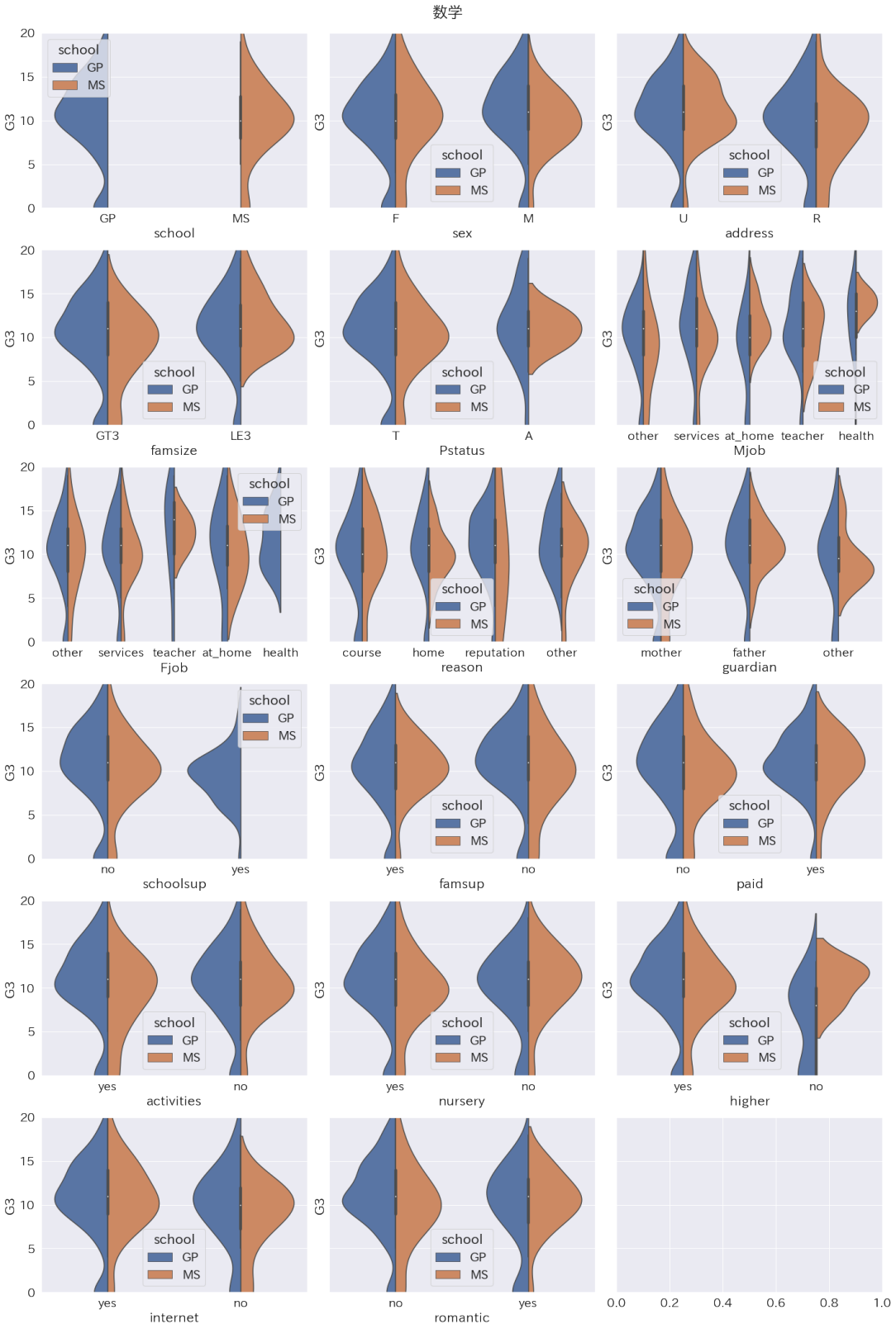

数学
count
school
GP
MS
sex
F
M
address
U
R
famsize
GT3
LE3
Pstatus
T
A
Mjob
other
services
at_home
teacher
health
Fjob
reason
course
home
reputation
guardian
mother
father
schoolsup
no
yes
famsup
paid
activities
nursery
higher
internet
romantic

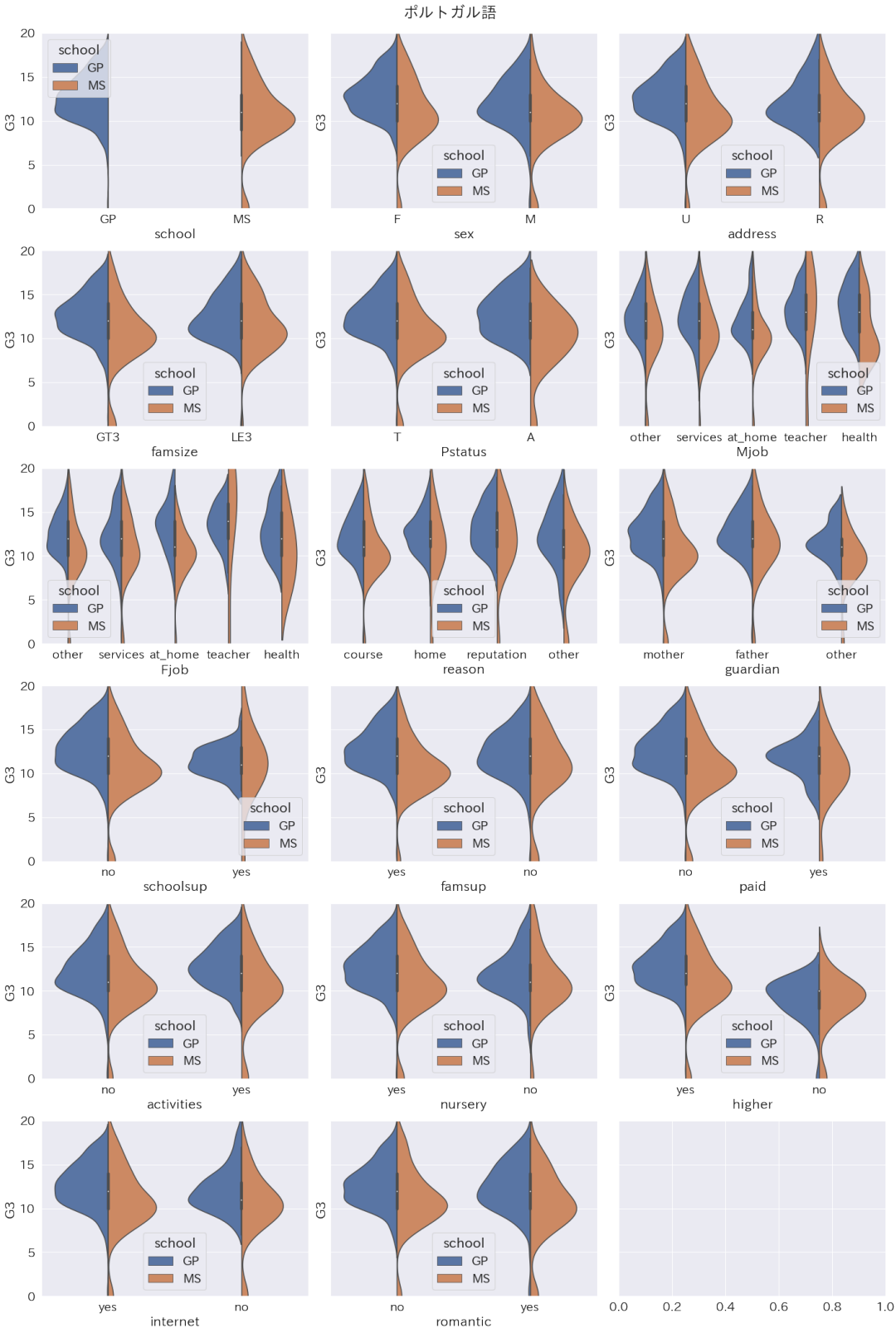


*数学とポルトガル語では paid の分布に特に大きな差

3.2 【作例】質的データに対する G3 の分布

[14]:
```
nor = len(cat)//3+1
noc = 3
for df1, sub in zip([df_m, df_p], ['数学', 'ポルトガル語']):
    fig, axs = plt.subplots(nor, noc, figsize=(20, 30), sharey=True)
    for i in range(nor):
        for j in range(noc):
            k = 3*i + j
            if k < len(cat):
                sns.violinplot(x=cat[k], y='G3', hue='school', split=True,␣
↪order=df1[cat[k]].value_counts().index, data=df1, ax=axs[i, j])
                axs[i, j].set_ylim([0, 20])
    fig.tight_layout()
    fig.suptitle(sub, fontsize=25)
    fig.subplots_adjust(top=0.96)
    plt.show()
```


数学
school
GP
MS
G3
0
5
10
15
20
GP
MS
school
F
M
sex
U
R
address
GT3
LE3
famsize
T
A
Pstatus
other
services
at_home
teacher
health
Mjob
Fjob
course
home
reputation
reason
mother
father
guardian
no
yes
schoolsup
famsup
paid
activities
nursery
higher
internet
romantic
0.0
0.2
0.4
0.6
0.8
1.0



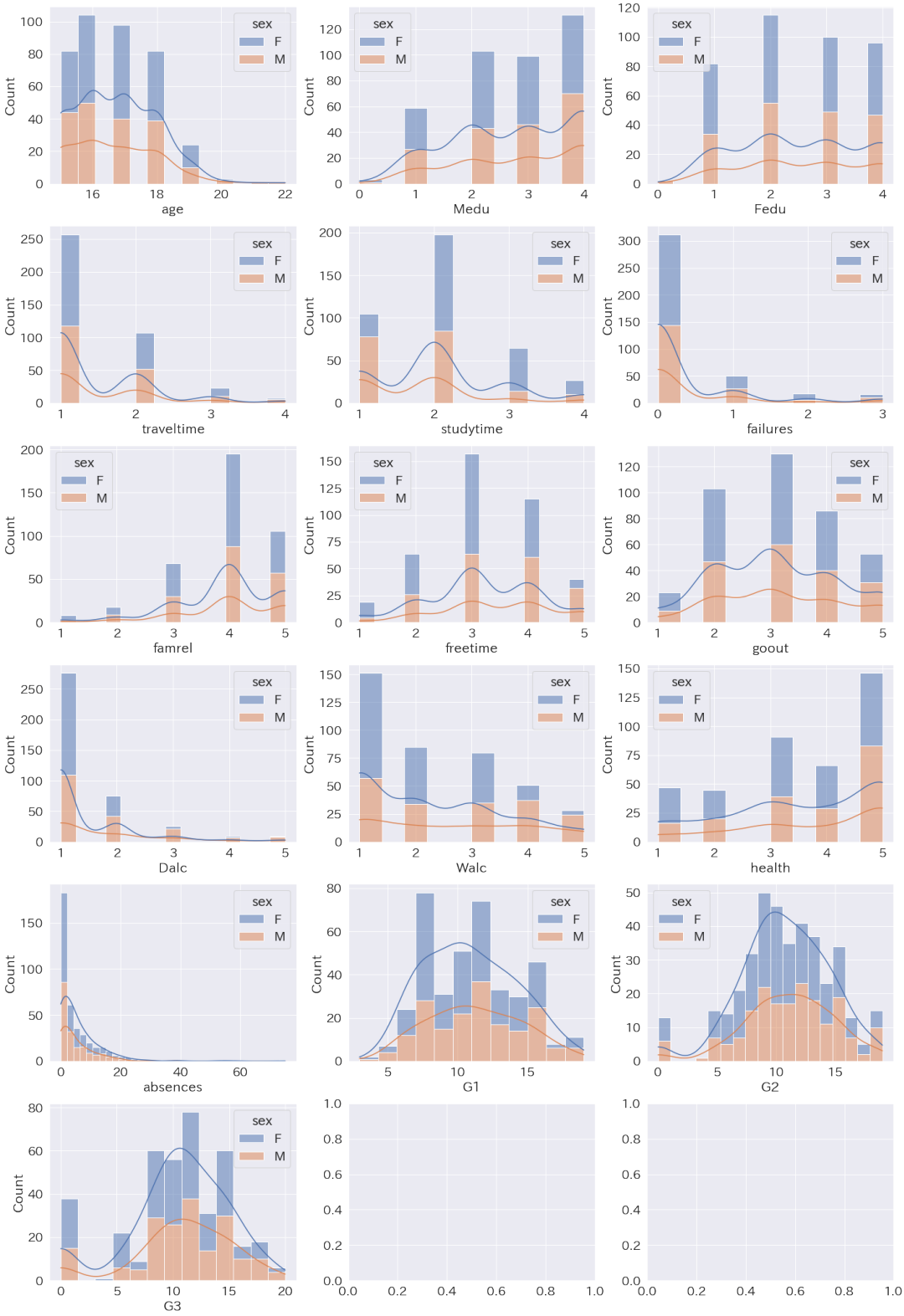
＊数学では guardian==other のピークが低い
＊ Mjob と Fjob は数学とポルトガル語で分布に違い

4　探索的分析：量的データ

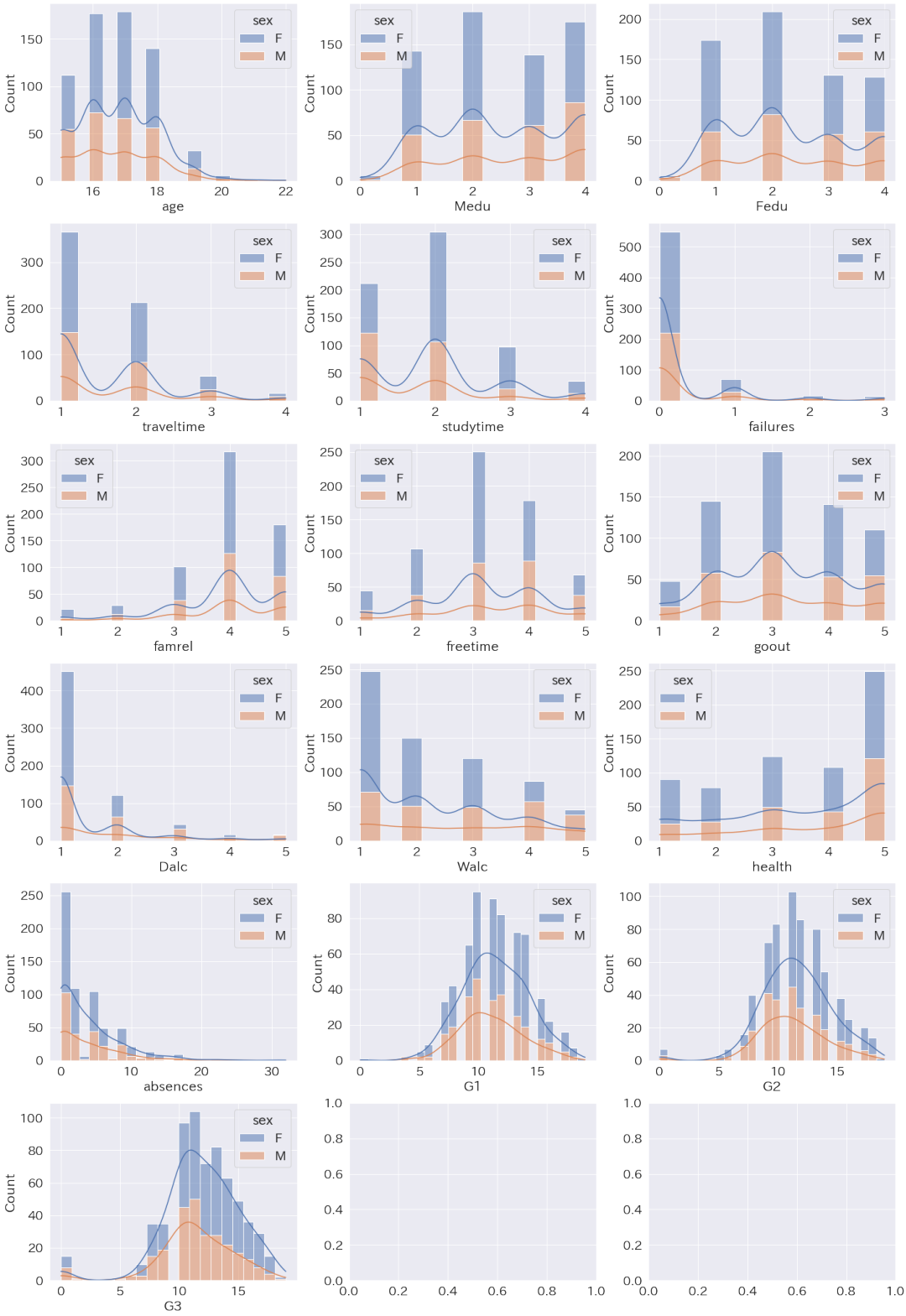
4.1　【作例】ヒストグラム

[15]:
```
nor = len(num)//3+1
noc = 3
for df1, sub in zip([df_m, df_p], ['数学', 'ポルトガル語']):
    fig, axs = plt.subplots(nor, noc, figsize=(20, 30))
    for i in range(nor):
        for j in range(noc):
            k = 3*i + j
            if k < len(num):
                sns.histplot(x=num[k], hue='sex', multiple='stack', kde=True, data=df1,␣
↪ax=axs[i, j])
    fig.tight_layout()
    fig.suptitle(sub, fontsize=25)
    fig.subplots_adjust(top=0.96)
    plt.show()
```

数学


sex
F
M
Count
age
Medu
Fedu
traveltime
studytime
failures
famrel
freetime
goout
Dalc
Walc
health
absences
G1
G2
G3

ポルトガル語

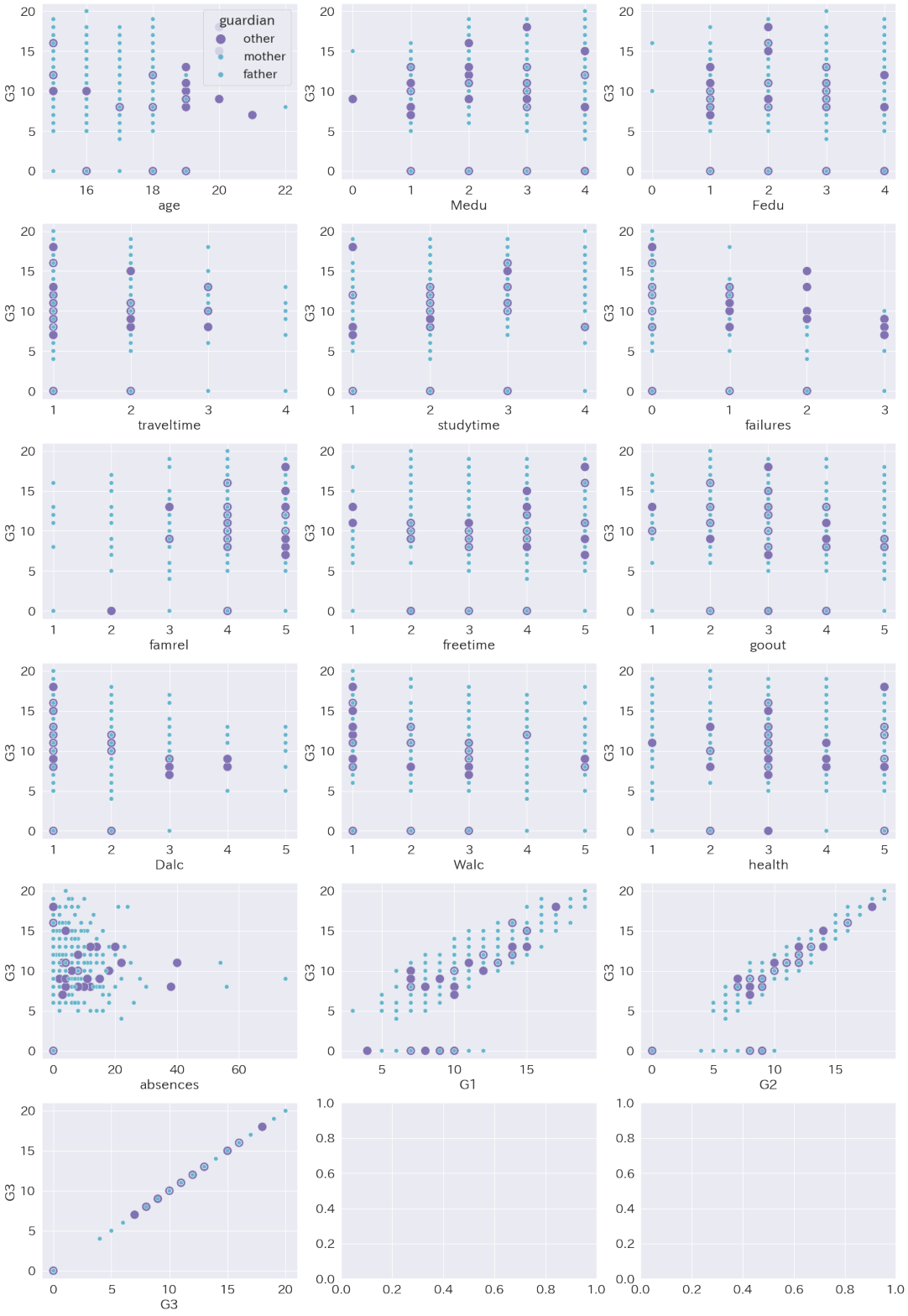


＊数学は成績に複数ピークが存在

4.2 【作例】散布図中における guradian==other の分布

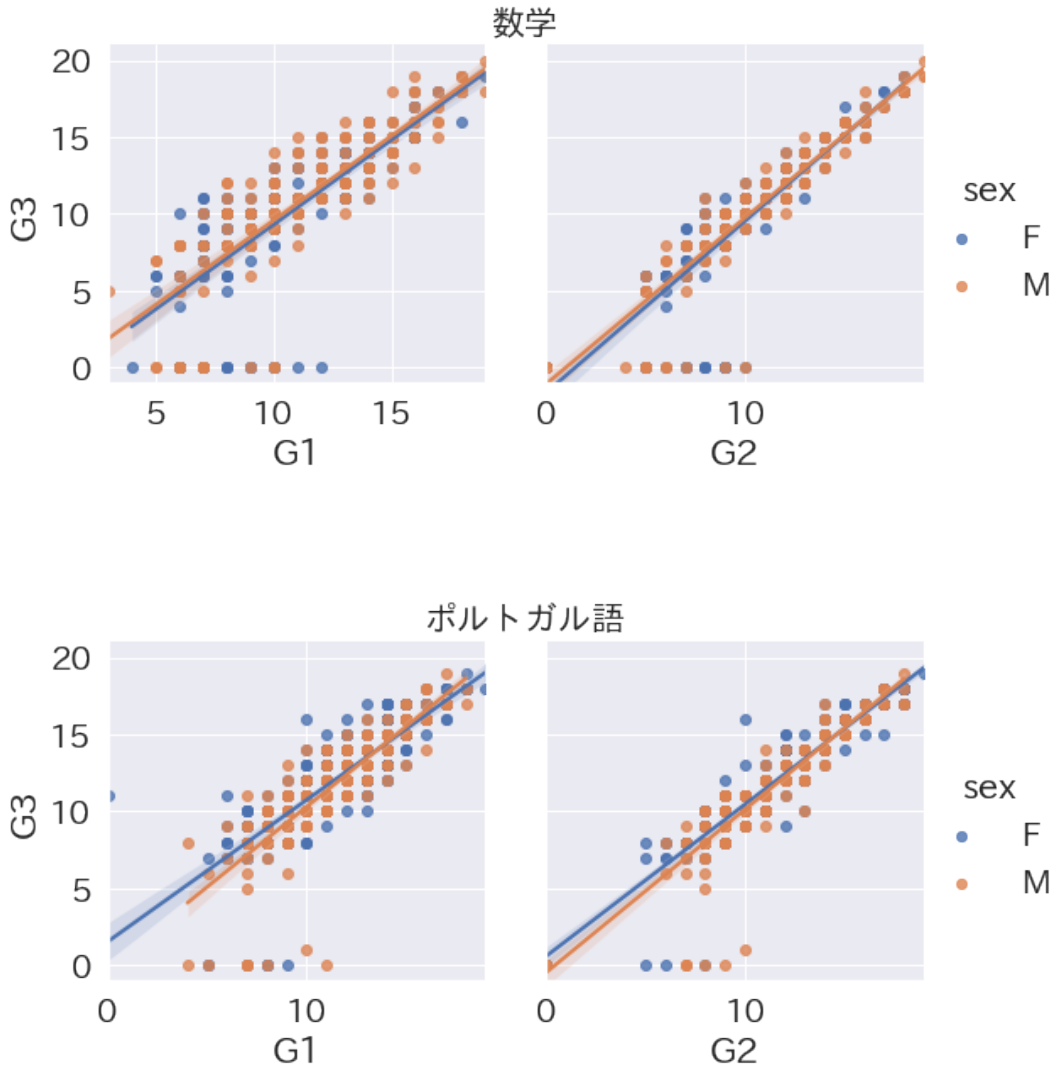
[16]:
```
targ = 'guardian'
myorder = ['other', 'mother', 'father']
mysizes = [200, 50, 50]
mypalette=['m','c','c']

nor = len(num)//3+1
noc = 3
fig, axs = plt.subplots(nor, noc, figsize=(20, 30))

for i in range(nor):
  for j in range(noc):
    k = 3*i + j
    if k < len(num):
      sns.scatterplot(x=num[k], y='G3', data=df_m, hue=targ, hue_order=myorder,␣
↪palette=mypalette, size=targ, size_order=myorder, sizes=mysizes, ax=axs[i, j])
      if k !=0:
        axs[i, j].get_legend().remove()

fig.suptitle('数学', fontsize=25)
fig.subplots_adjust(top=0.4)
fig.tight_layout()
```

数学


guardian
other
mother
father
G3
age
Medu
Fedu
traveltime
studytime
failures
famrel
freetime
goout
Dalc
Walc
health
absences
G1
G2
G3

4.3 【作例】学期成績と学年末成績の相関

[17]:
```
for df1, sub in zip([df_m, df_p], [' 数学', ' ポルトガル語']):
    g = sns.PairGrid(df1, y_vars=["G3"], x_vars=["G1", "G2"], height=4, hue='sex')
    g.map(sns.regplot)
    g.set(ylim=(-1, 21))
    g.fig.subplots_adjust(top=0.9)
    g.fig.suptitle(sub, fontsize=20)
    g.add_legend()
    plt.show()
```

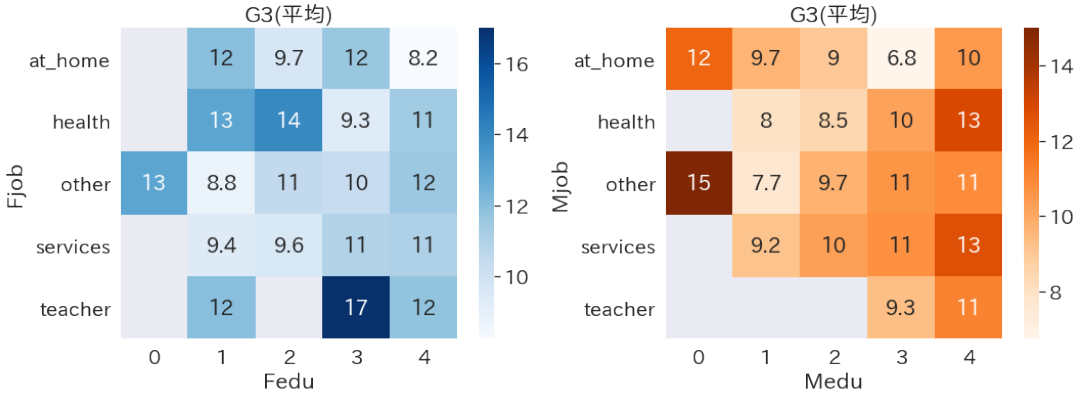


4.4 【作例】両親の学歴・職業と学年末成績（平均）

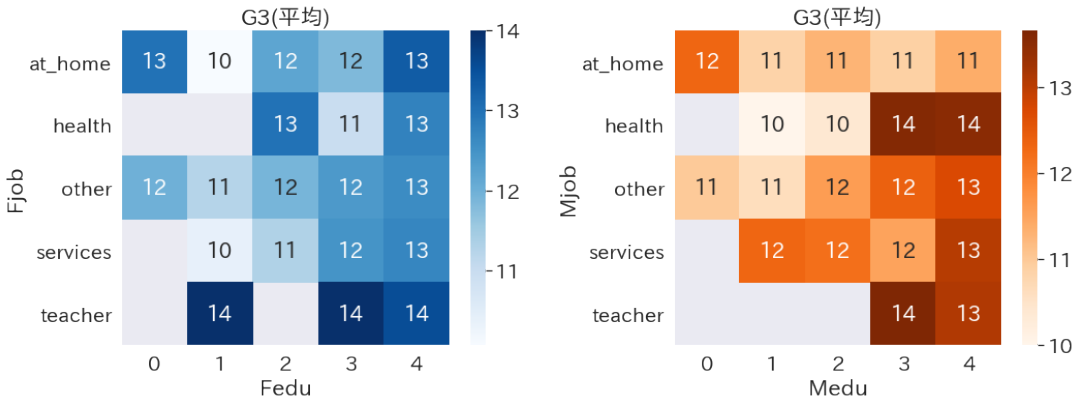
[18]:
```
def groupmean(df1, col1, col2, col3):

  # df1 に対して col1 と col2 で groupby して col3 の平均を取得
  df_gb = df1.groupby([col1, col2], as_index=False)[col3].mean()

  # 出力 df の外枠作成
  df_out = pd.DataFrame(
    index = df_gb[col1].unique(),
    columns= df_gb[col2].sort_values().unique())

  # 出力 df 要素の配置
  for i in range(df_gb.shape[0]):
    v = df_gb.iloc[i]
    df_out.loc[v[0], v[1]] = v[2]
  return df_out
```

[19]:
```
mp = [df_m, df_p]
dm = [['Fjob', 'Fedu', 'G3'], ['Mjob', 'Medu', 'G3']]
st = ['数学', 'ポルトガル語']
cm = ['Blues', 'Oranges']

# Math or Portuguese
for i in range(2):
  fig, axs = plt.subplots(1, 2, figsize=(15, 6))

  # Dad or Mom
  for j in range(2):
    df_out = groupmean(mp[i], *dm[j])
    sns.heatmap(df_out.fillna(0), mask=df_out.isna(), cmap=cm[j], annot=True,␣
↪annot_kws={"size": 20}, ax=axs[j])
    axs[j].set(title='G3(平均)', xlabel=dm[j][1], ylabel=dm[j][0])
  fig.suptitle(st[i], fontsize=20)
  fig.subplots_adjust(top=1)
  fig.tight_layout()
  plt.show()
```

数学

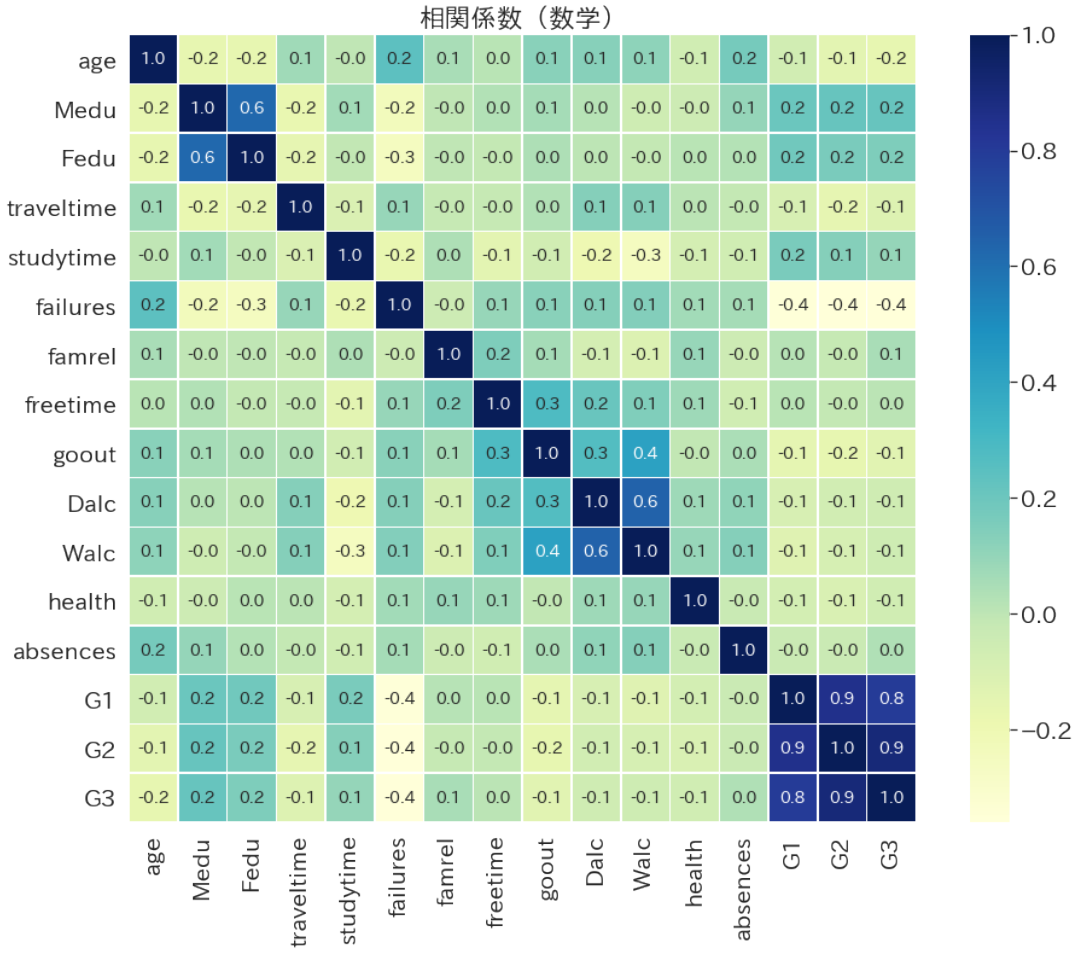

G3(平均)
at_home
health
other
services
teacher
Fjob
12 9.7 12 8.2
13 14 9.3 11
13 8.8 11 10 12
9.4 9.6 11 11
12 17 12
0 1 2 3 4
Fedu
16 14 12 10
G3(平均)
at_home
health
other
services
teacher
Mjob
12 9.7 9 6.8 10
8 8.5 10 13
15 7.7 9.7 11 11
9.2 10 11 13
9.3 11
0 1 2 3 4
Medu
14 12 10 8

ポルトガル語


G3(平均)
at_home
health
other
services
teacher
Fjob
13 10 12 12 13
13 11 13
12 11 12 12 13
10 11 12 13
14 14 14
0 1 2 3 4
Fedu
14 13 12 11
G3(平均)
at_home
health
other
services
teacher
Mjob
12 11 11 11 11
10 10 14 14
11 11 12 12 13
12 12 12 13
14 13
0 1 2 3 4
Medu
13 12 11 10

4.5 【作例】相関係数

[20]:
```
corr = df_m.corr()
plt.subplots(figsize=(16, 12))
sns.heatmap(corr, square=True, linewidth=.5, annot=True, annot_kws={"size": 14},␣
↪cmap='YlGnBu', fmt='.1f')
plt.title('相関係数（数学）')
plt.show()
```



5　機械学習による成績予測

5.1　【作例】線形回帰

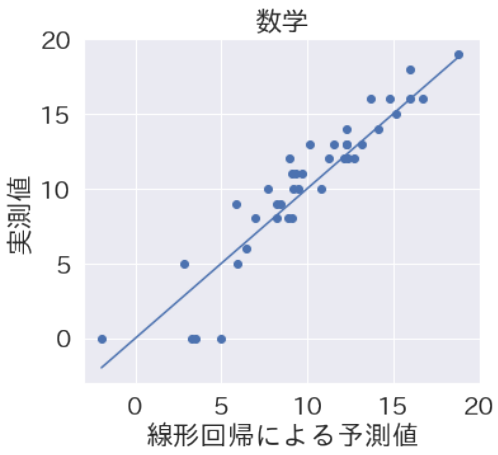
[21]:
```
df_rmse = pd.DataFrame(
    index=['訓練データ', 'テストデータ', '全データ'],
    columns=['数学', 'ポルトガル語', '参考値'])
df_rmse.iloc[2, 2] = 1.32
```

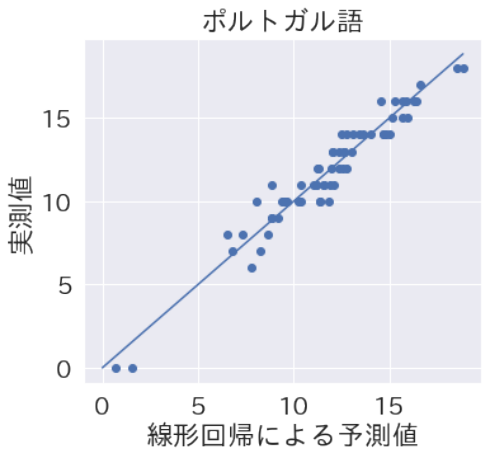
[22]:
```
# 標準化
def zval(df1):
    return (df1-df1.mean())/df1.std(ddof=1)
z = zval(df_m[num])

# onehot ベクトル化
w = pd.get_dummies(df_m[cat], drop_first=True)

x = z.join(w)
y = x['G3']
x = x.drop(columns=['G3'])

# データ分割
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.1, random_state=123)

# 線形回帰モデル
from sklearn.linear_model import LinearRegression as LR
lr = LR()
lr.fit(x_train, y_train)
y_train_pred = lr.predict(x_train)
y_test_pred = lr.predict(x_test)

# 標準化の逆変換（G3用）
def zinv(x):
    return x * df_m['G3'].std(ddof=1) + df_m['G3'].mean()

y_train_pred = zinv(y_train_pred)
y_test_pred = zinv(y_test_pred)
y_train = zinv(y_train)
y_test = zinv(y_test)

from sklearn.metrics import mean_squared_error as MSE
rmse_train = np.sqrt(MSE(y_train, y_train_pred))
rmse_test = np.sqrt(MSE(y_test, y_test_pred))
df_rmse['数学'] = [rmse_train, rmse_test, np.sqrt((rmse_train**2+rmse_test**2)/2)]

fig, axs=plt.subplots(1, 2, figsize=(14,5), gridspec_kw=dict(wspace=.5))
```

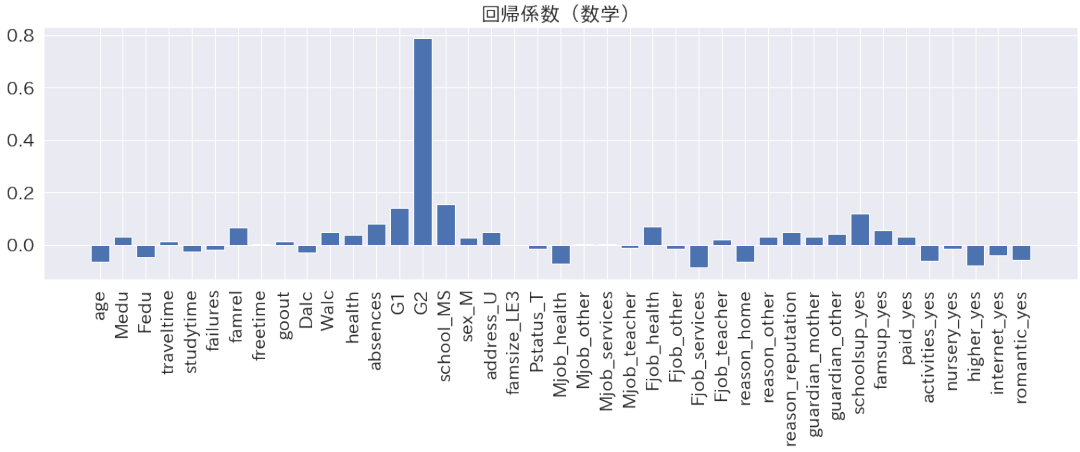
```
axs[0].scatter(y_test_pred, y_test)
max_value = np.maximum(np.max(y_test), np.max(y_test_pred))
min_value = np.minimum(np.min(y_test), np.min(y_test_pred))
axs[0].plot([min_value, max_value],[min_value,max_value])
axs[0].set_xlabel('線形回帰による予測値')
axs[0].set_ylabel('実測値')
axs[0].set_title('数学')
import pickle
with open("./math_reg.pkl", "wb") as f:
    pickle.dump((fig, axs), f)

fig = plt.figure(figsize=(20,5))
plt.bar(x=list(x_train.columns), height=lr.coef_)
plt.xticks(rotation=90)
plt.title('回帰係数（数学）')
with open("./math_coe.pkl", "wb") as f:
    pickle.dump(fig, f)
#plt.show()
```

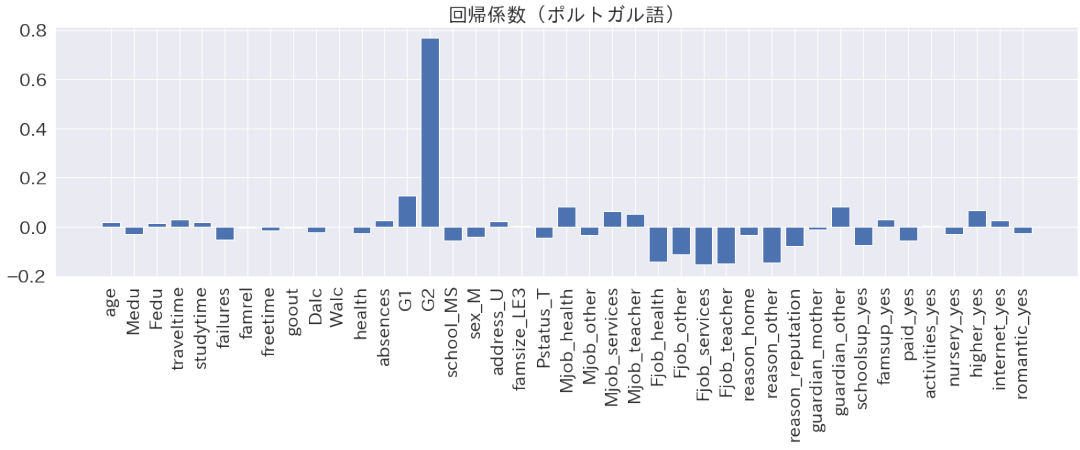
[23]:
```
# 標準化
def zval(x):
    return (x-x.mean())/x.std(ddof=1)
z = zval(df_p[num])

# onehot ベクトル化
w = pd.get_dummies(df_p[cat], drop_first=True)

x = z.join(w)
y = x['G3']
x = x.drop(columns=['G3'])

# データ分割
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.1, random_state=123)

# 線形回帰モデル
from sklearn.linear_model import LinearRegression as LR
lr = LR()
lr.fit(x_train, y_train)
y_train_pred = lr.predict(x_train)
y_test_pred = lr.predict(x_test)

# 標準化の逆変換
def zinv(x):
    return x * df_p['G3'].std(ddof=1) + df_p['G3'].mean()

y_train_pred = zinv(y_train_pred)
```

```
y_test_pred = zinv(y_test_pred)
y_train = zinv(y_train)
y_test = zinv(y_test)

from sklearn.metrics import mean_squared_error as MSE
rmse_train = np.sqrt(MSE(y_train, y_train_pred))
rmse_test = np.sqrt(MSE(y_test, y_test_pred))
df_rmse[' ポルトガル語'] = [rmse_train, rmse_test, np.sqrt((rmse_train**2+rmse_test**2)/2)]

with open("./math_reg.pkl", "rb") as f:
    fig, axs=pickle.load(f)

axs[1].scatter(y_test_pred, y_test)
max_value = np.maximum(np.max(y_test), np.max(y_test_pred))
min_value = np.minimum(np.min(y_test), np.min(y_test_pred))
axs[1].plot([min_value, max_value],[min_value,max_value])
axs[1].set_xlabel(' 線形回帰による予測値')
axs[1].set_ylabel(' 実測値')
axs[1].set_title(' ポルトガル語')
plt.show()

with open("./math_coe.pkl", "rb") as f:
    fig=pickle.load(f)
plt.show()



plt.figure(figsize=(20,5))
plt.bar(x=list(x_train.columns), height=lr.coef_)
plt.xticks(rotation=90)
plt.title(' 回帰係数（ポルトガル語）')
plt.show()
```





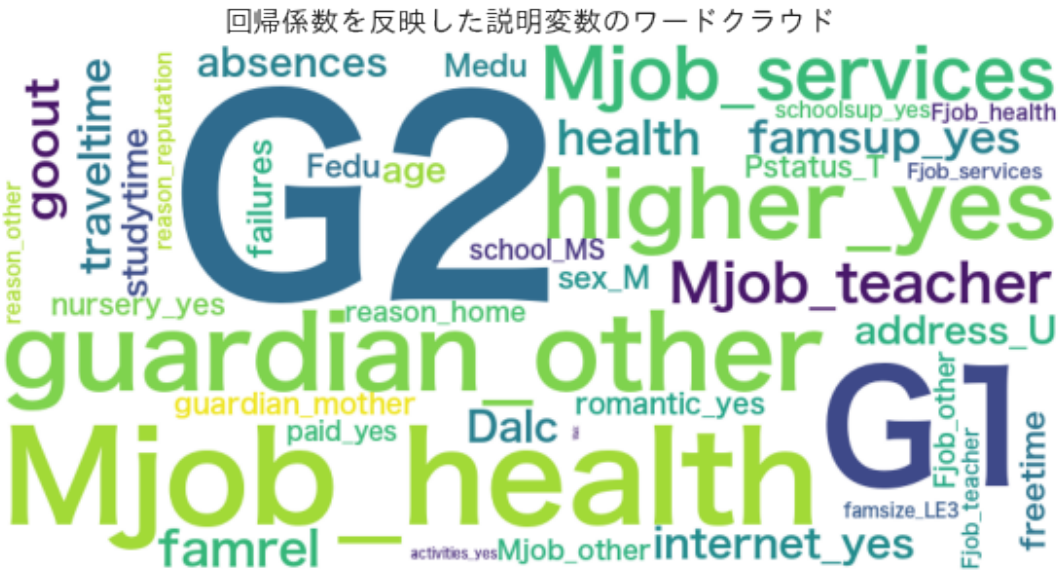




二乗平均平方根誤差

	数学	ポルトガル語	参考値 [2]
訓練データ	1.82	1.24	NaN
テストデータ	1.75	0.86	NaN
全データ	1.78	1.07	1.32

[25]:
```
from wordcloud import WordCloud

wcdic = {x: y for x, y in zip(x_train.columns, lr.coef_)}

wc = WordCloud(
    font_path="/System/Library/Fonts/ヒラギノ角ゴシック W6.ttc",
    background_color='white',
    width=800, height=400)
wc.generate_from_frequencies(wcdic)

plt.figure(figsize=(20,10))
plt.title('回帰係数を反映した説明変数のワードクラウド', fontsize=15)
plt.imshow(wc)
plt.axis("off")
plt.show()
```



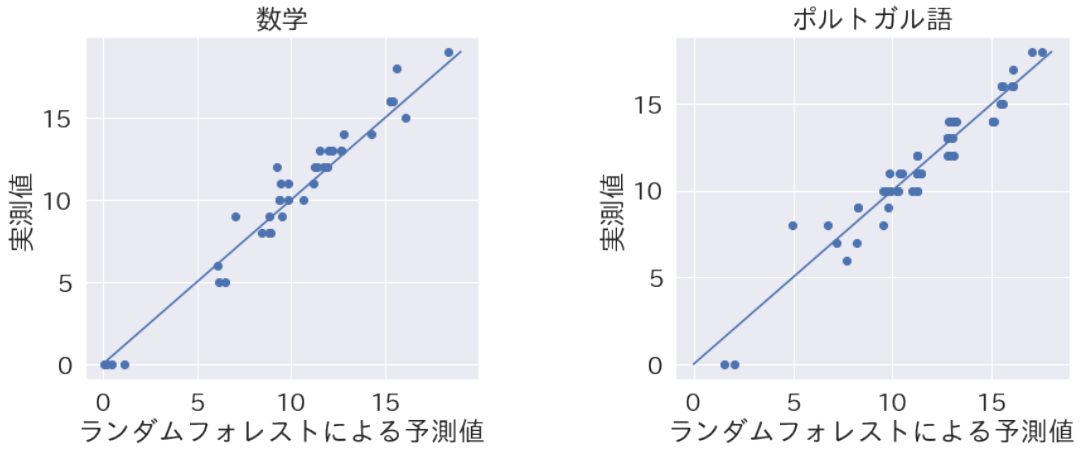
5.2 【作例】ランダムフォレスト

[25]:
```
# 標準化
def zval(x):
    return (x-x.mean())/x.std(ddof=1)
z = zval(df_m[num])

# onehot ベクトル化
w = pd.get_dummies(df_m[cat], drop_first=True)

x = z.join(w)
y = x['G3']
x = x.drop(columns=['G3'])

# データ分割
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.1, random_state=123)
# print(df_m.shape, x_train.shape[0], x_test.shape[0])

# ランダムフォレスト
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(max_depth=8, random_state=123)
rf.fit(x_train, y_train)
y_train_pred = rf.predict(x_train)
y_test_pred = rf.predict(x_test)

# 標準化の逆変換
def zinv(x):
    return x * df_m['G3'].std(ddof=1) + df_m['G3'].mean()

y_train_pred = zinv(y_train_pred)
y_test_pred = zinv(y_test_pred)
y_train = zinv(y_train)
y_test = zinv(y_test)

from sklearn.metrics import mean_squared_error as MSE
rmse_train = np.sqrt(MSE(y_train, y_train_pred))
rmse_test = np.sqrt(MSE(y_test, y_test_pred))
df_rmse[' 数学'] = [rmse_train, rmse_test, np.sqrt((rmse_train**2+rmse_test**2)/2)]

fig, axs=plt.subplots(1, 2, figsize=(14,5), gridspec_kw=dict(wspace=.5))
axs[0].scatter(y_test_pred, y_test)
max_value = np.maximum(np.max(y_test), np.max(y_test_pred))
min_value = np.minimum(np.min(y_test), np.min(y_test_pred))
axs[0].plot([min_value, max_value],[min_value,max_value])
axs[0].set_xlabel(' ランダムフォレストによる予測値')
```

```
axs[0].set_ylabel('実測値')
axs[0].set_title('数学')
import pickle
with open("./math_reg.pkl", "wb") as f:
    pickle.dump((fig, axs), f)
```

[26]:
```
# 標準化
def zval(x):
    return (x-x.mean())/x.std(ddof=1)
z = zval(df_p[num])

# onehot ベクトル化
w = pd.get_dummies(df_p[cat], drop_first=True)

x = z.join(w)
y = x['G3']
x = x.drop(columns=['G3'])

# データ分割
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.1, random_state=123)
# print(df_p.shape, x_train.shape[0], x_test.shape[0])

# ランダムフォレスト
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(max_depth=4, random_state=123)
rf.fit(x_train, y_train)
y_train_pred = rf.predict(x_train)
y_test_pred = rf.predict(x_test)

# 標準化の逆変換
def zinv(x):
    return x * df_p['G3'].std(ddof=1) + df_p['G3'].mean()

y_train_pred = zinv(y_train_pred)
y_test_pred = zinv(y_test_pred)
y_train = zinv(y_train)
y_test = zinv(y_test)

from sklearn.metrics import mean_squared_error as MSE
rmse_train = np.sqrt(MSE(y_train, y_train_pred))
rmse_test = np.sqrt(MSE(y_test, y_test_pred))
df_rmse['ポルトガル語'] = [rmse_train, rmse_test, np.sqrt((rmse_train**2+rmse_test**2)/2)]



with open("./math_reg.pkl", "rb") as f:
    fig, axs=pickle.load(f)
```

```
axs[1].scatter(y_test_pred, y_test)
max_value = np.maximum(np.max(y_test), np.max(y_test_pred))
min_value = np.minimum(np.min(y_test), np.min(y_test_pred))
axs[1].plot([min_value, max_value],[min_value,max_value])
axs[1].set_xlabel(' ランダムフォレストによる予測値')
axs[1].set_ylabel(' 実測値')
axs[1].set_title(' ポルトガル語')
plt.show()
```



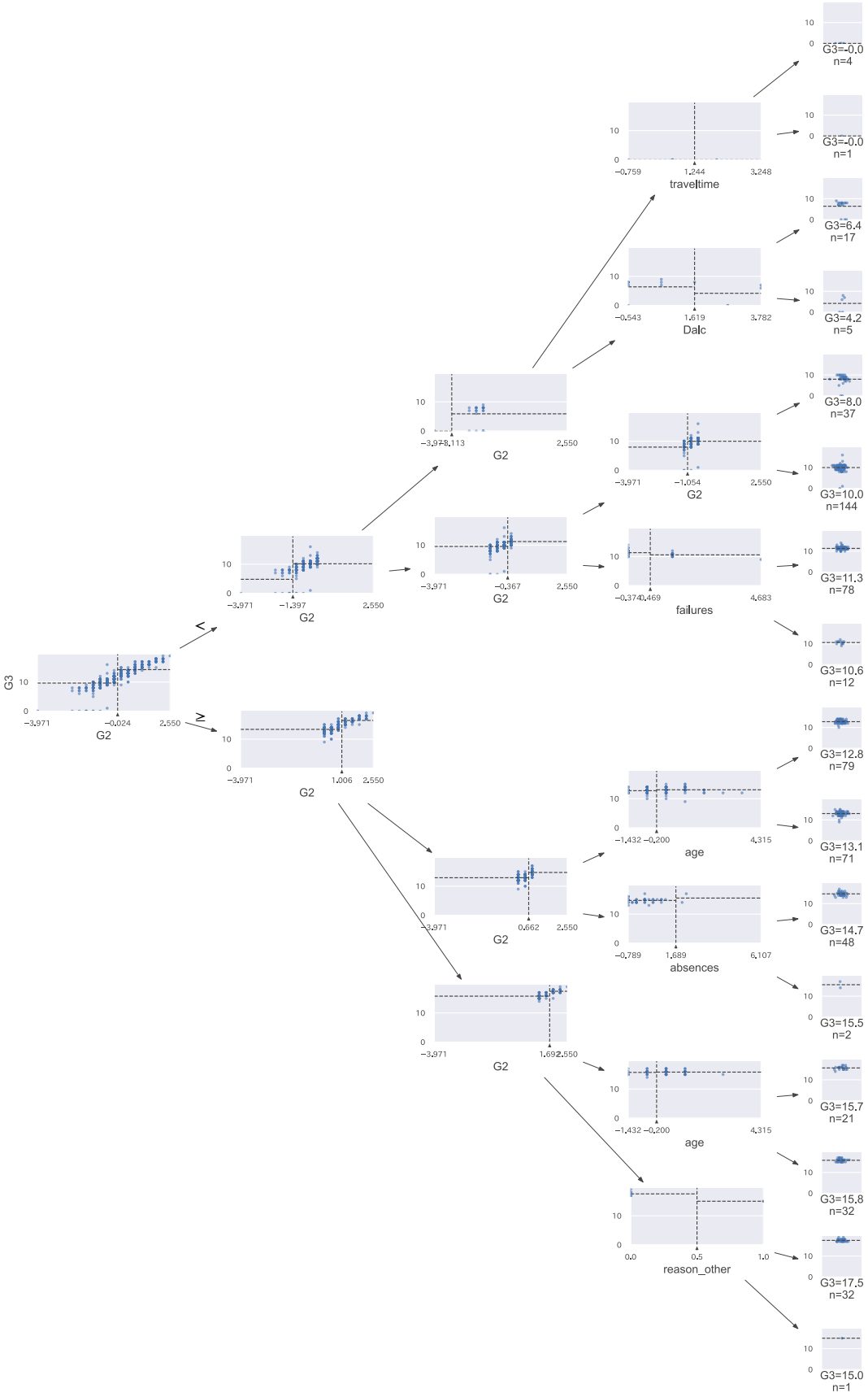
二乗平均平方根誤差

	数学	ポルトガル語	参考値 [2]
訓練データ	0.64	1.07	NaN
テストデータ	1.01	0.89	NaN
全データ	0.84	0.98	1.32

[28]:
```
from dtreeviz.trees import dtreeviz
estimators = rf.estimators_
viz = dtreeviz(
    estimators[0],
    x_train,
    y_train,
    target_name='G3',
    feature_names=x.columns.values,
    precision=1,
    orientation='LR',
    title=' 決定木の可視化（ポルトガル語）'
)

# viz
```

[29]:

決定木の可視化 (ポルトガル語)



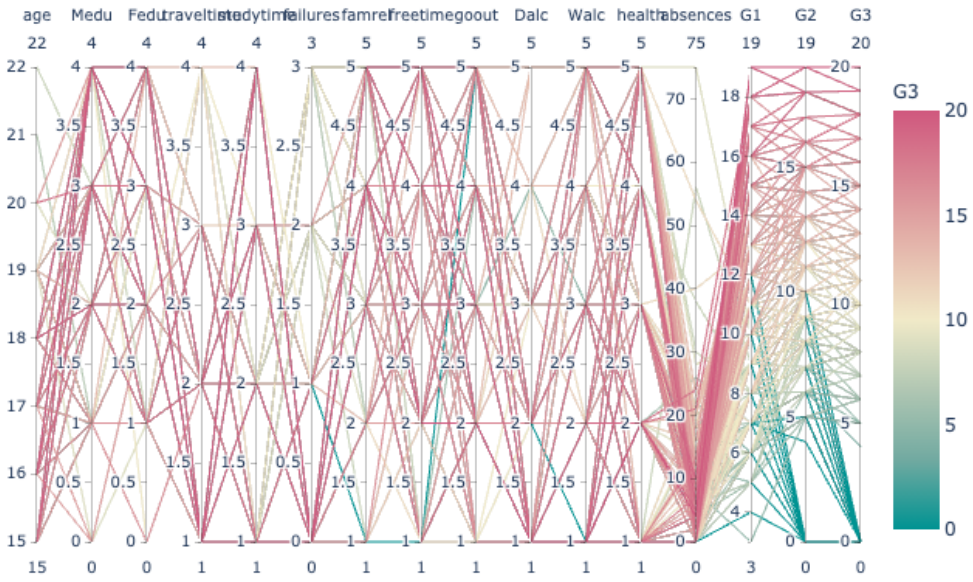
6　その他

6.1　G2==0 と G3==0 について

以下を考慮すると、成績カラムの値 0 は実際には NaN（何らかの理由で該当科目を履修しなくなった）であると考えられる。

- 二学期以降、成績に 1 から 3 が存在しないこと
- 成績に 0 がついていない生徒に関しては、次学期に点数が下がった場合、その差は最大でも-4 点であること
- 成績に 0 がついた生徒に関しては、次学期に点数が下がった場合、その差は最大で-12 点であること

[30]:
```
import plotly.express as px
fig = px.parallel_coordinates(df_m, dimensions=num, color='G3', color_continuous_scale=px.
 ↪colors.diverging.Tealrose)
fig.show()
```



[31]:
```
df_m['dG12'] = df_m['G2'] - df_m['G1']
df_m['dG23'] = df_m['G3'] - df_m['G2']
df_m[((df_m['G2']!=0) & (df_m['G3']!=0)) & ((df_m['dG12']<-2) | (df_m['dG23']<-1))].
 ↪sort_values('dG12')[['G1', 'G2', 'G3', 'dG12', 'dG23']]
```

[31]:

	G1	G2	G3	dG12	dG23
156	16	12	13	-4	1
123	14	11	13	-3	2
155	11	8	8	-3	0
302	15	12	14	-3	2
308	15	12	12	-3	0

313	13	10	11	-3	1
45	8	8	6	0	-2
100	7	7	5	0	-2
176	13	13	11	0	-2
216	6	6	4	0	-2
174	10	11	9	1	-2
393	11	12	10	1	-2
161	5	9	7	4	-2

参考文献

[1] カリフォルニア大学アーバイン校／機械学習リポジトリ
http://archive.ics.uci.edu/ml/datasets/Student+Performance

[2] USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE
http://www3.dsi.uminho.pt/pcortez/student.pdf