

● ชิกญี่ปุ่น ✓



# Human Resource

BASIC PROGRAMMING FOR DATA SCIENTISTS  
AND DATA VISUALIZATION

# Members



5 ณัชชา ประทุมชาติ



12 พรรณรณมณ ราชคมนตรี



15 รัชชานนท์ พันภาพสินธุ์



27 อริสรา ยังอยู่

# Get to know data

- **enrollee\_id** : รหัสของผู้ลงทะเบียน
- **city** : รหัสเมือง
- **citydevelopmentindex** : ดัชนีการพัฒนามืองเมือง (ปรับมาตรฐานแล้ว)
- **gender** : เพศของผู้สมัคร
- **relevent\_experience** : ประสบการณ์ที่เกี่ยวข้องของผู้สมัคร
- **enrolled\_university** : ประเภทของหลักสูตรมหาวิทยาลัยที่กำลังศึกษาอยู่ (ถ้ามี)
- **education\_level** : ระดับการศึกษาของผู้ลงทะเบียน
- **major\_discipline** : สาขาวิชาหลักที่ผู้สมัครจบการศึกษา
- **experience** : ประสบการณ์ทำงานทั้งหมดของผู้สมัคร (หน่วยปี)
- **company\_size** : จำนวนพนักงานในบริษัทที่ทำงานปัจจุบัน
- **company\_type** : ประเภทของบริษัทที่ทำงานปัจจุบัน
- **lastnewjob** : ความแตกต่างในจำนวนปีระหว่างงานก่อนหน้าและงานปัจจุบัน
- **training\_hours** : จำนวนชั่วโมงการฝึกอบรมที่เสร็จสิ้น
- **target** : 0 - ไม่ต้องการเปลี่ยนงาน  
1 - กำลังมองหางานใหม่

# Extract Data

```
import pandas as pd
import os

path_to_hr_ov = '/content/drive/MyDrive/data_viz_2024_DATA/project/hr_overview.csv'
```

```
hr_ov = pd.read_csv(path_to_hr_ov)
hr_ov
```

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours	target
0	8949	103	0.920	Male	Yes	no_enrollment	Graduate	STEM	>20	unknown	unknown	1	36	1.0
1	29725	40	0.776	Male	No	no_enrollment	Graduate	STEM	15	50-99	Pvt Ltd	>4	47	0.0
2	11561	21	0.624	unknown	No	Full time course	Graduate	STEM	5	unknown	unknown	never	83	0.0
3	33241	115	0.789	unknown	No	no_enrollment	Graduate	Business Degree	<1	unknown	Pvt Ltd	never	52	1.0
4	666	162	0.767	Male	Yes	no_enrollment	Masters	STEM	>20	50-99	Funded Startup	4	8	0.0

# Extract Data

- education\_score

```
# ฟังก์ชันคำนวณคะแนนการศึกษา โดยใช้ weight
def calculate_weighted_education_score(row):
    score = 0

    # น้ำหนักสำหรับ education_level
    education_weights = {
        'Primary School': 1,
        'High School': 3,
        'Graduate': 5,
        'Masters': 7,
        'PhD': 9
    }
    score += education_weights.get(row['education_level'], 0) * 1.5 # เพิ่มน้ำหนัก

    # น้ำหนักสำหรับ major_discipline
    discipline_weights = {
        'STEM': 3,
        'Humanities': 1,
        'Other': 1,
        'Business Degree': 2,
        'Arts': 1
    }
    score += discipline_weights.get(row['major_discipline'], 0) * 1.2 # เพิ่มน้ำหนัก
```

```
# น้ำหนักสำหรับ enrolled_university
enrollment_weights = {
    'no_enrollment': 0,
    'Part time course': 2,
    'Full time course': 3
}
score += enrollment_weights.get(row['enrolled_university'], 0) * 1.0 # เพิ่มน้ำหนัก

return score # คืนคะแนนรวม

# คำนวณคะแนนการศึกษา
hr_ov['education_score'] = hr_ov.apply(calculate_weighted_education_score, axis=1)

# Normalization โดยให้คะแนนสูงสุดอยู่ที่ 10
max_score = hr_ov['education_score'].max()
hr_ov['education_score'] = (hr_ov['education_score'] / max_score) * 10

# แสดงผลลัพธ์
print(hr_ov[['education_score']].head())
```

	education_score
0	6.491228
1	6.491228
2	8.245614
3	5.789474
4	8.245614

# Extract Data

## • experience\_score

```
import pandas as pd

def calculate_weighted_experience_score(row):
    """Calculates experience score based on experience and relevant experience."""
    # ตรวจสอบว่า experience เป็น 'unknown'
    if row['experience'] == 'unknown':
        return 'unknown' # คืนค่า 'unknown' ถ้าข้อมูลเป็น 'unknown'

    score = 0

    try:
        # แปลงค่า experience เป็น int
        experience = int(row['experience'])
        # คำนวณคะแนนตามช่วงของประสบการณ์
        if experience <= 5:
            score += experience * 1.5
        elif experience > 5 and experience <= 10:
            score += 5 * 1.5 + (experience - 5) * 1
        elif experience > 10 and experience <= 20:
            score += 5 * 1.5 + 5 * 1 + (experience - 10) * 0.5
        elif experience > 20:
            score += 5 * 1.5 + 5 * 1 + 10 * 0.5 + (experience - 20) * 0.25
    except ValueError:
        pass

    # Normalize คะแนนให้เต็ม 10
    # คะแนนสูงสุดในกรณีที่ มีประสบการณ์ 20+ และมี relevant_experience คือ 10
    max_score = 10
    normalized_score = (score / max_score) * 10

    return min(normalized_score, 10) # คืนคะแนนไม่เกิน 10

# คำนวณคะแนนการประสบการณ์
hr_ov['experience_score'] = hr_ov.apply(calculate_weighted_experience_score, axis=1)

# ตรวจสอบค่าที่ได้ในคอลัมน์ experience_score
print(hr_ov[['experience', 'experience_score']].head())
```

```
# เพิ่มคะแนนสำหรับ relevant_experience
if row['relevant_experience'] == 'Yes':
    score += 2
except ValueError:
    if row['experience'] == '<1':
        score += 1
    elif row['experience'] == '>20':
        score += 10
    else:
        pass

# Normalize คะแนนให้เต็ม 10
# คะแนนสูงสุดในกรณีที่ มีประสบการณ์ 20+ และมี relevant_experience คือ 10
max_score = 10
normalized_score = (score / max_score) * 10

return min(normalized_score, 10) # คืนคะแนนไม่เกิน 10

# คำนวณคะแนนการประสบการณ์
hr_ov['experience_score'] = hr_ov.apply(calculate_weighted_experience_score, axis=1)

# ตรวจสอบค่าที่ได้ในคอลัมน์ experience_score
print(hr_ov[['experience', 'experience_score']].head())
```

	experience	experience_score
0	>20	10.0
1	15	10.0
2	5	7.5
3	<1	1.0
4	>20	10.0



# Extract Data

- **training\_score**

```
def calculate_training_score(row):  
    """Calculates training score based on training_hours."""  
    training_hours = row['training_hours']  
  
    # ตรวจสอบว่ามีข้อมูล training_hours หรือไม่  
    if pd.isnull(training_hours):  
        return 'unknown' # คืนค่า 'unknown' หากไม่มีข้อมูล  
  
    max_hours = hr_ov['training_hours'].max() # ชั่วโมงสูงสุดใน training_hours  
  
    # คำนวณคะแนนการฝึกอบรม  
    score = (training_hours / max_hours) * 10  
  
    return min(score, 10) # Ensure score doesn't exceed 10  
  
# คำนวณคะแนนการฝึกอบรม  
hr_ov['training_score'] = hr_ov.apply(calculate_training_score, axis=1)  
  
# ตรวจสอบค่าที่ได้ในคอลัมน์ training_score  
print(hr_ov[['training_hours', 'training_score']].head())
```

	training_hours	training_score
0	36	1.071429
1	47	1.398810
2	83	2.470238
3	52	1.547619
4	8	0.238095

# Extract Data

- company\_exposure

```

4 def calculate_company_exposure(row):
5     """Calculates company exposure score based on company_size and company_type."""
6     # ตรวจสอบข้อมูลที่เป็น 'unknown'
7     if row['company_size'] == 'unknown' or row['company_type'] == 'unknown':
8         return 'unknown' # คืนค่า 'unknown' หากข้อมูลเป็น 'unknown'
9
10    score = 0
11    weight_company_size = 0.6 # น้ำหนักให้กับ company_size
12    weight_company_type = 0.4 # น้ำหนักให้กับ company_type
13
14    # คะแนนตามขนาดบริษัท
15    if row['company_size'] == '<10':
16        score += 2 * weight_company_size
17    elif row['company_size'] == '10-49':
18        score += 3 * weight_company_size
19    elif row['company_size'] == '50-99':
20        score += 4 * weight_company_size
21    elif row['company_size'] == '100-500':
22        score += 5 * weight_company_size
23    elif row['company_size'] == '500-999':
24        score += 6 * weight_company_size
25    elif row['company_size'] == '1000-4999':
26        score += 7 * weight_company_size
27    elif row['company_size'] == '5000-9999':
28        score += 8 * weight_company_size
29    elif row['company_size'] == '10000+':
30        score += 9 * weight_company_size
31
32    # คะแนนตามประเภทบริษัท
33    if row['company_type'] == 'Pvt Ltd':
34        score += 4 * weight_company_type
35    elif row['company_type'] == 'Funded Startup':
36        score += 6 * weight_company_type
37    elif row['company_type'] == 'Public Sector':
38        score += 3 * weight_company_type
39    elif row['company_type'] == 'NGO':
40        score += 2 * weight_company_type
41    elif row['company_type'] == 'Early Stage Startup':
42        score += 5 * weight_company_type
43    elif row['company_type'] == 'Other':
44        score += 1 * weight_company_type
45
46    return score # คืนคะแนนรวม

```

```

48 # คำนวณคะแนน
49 hr_ov['company_exposure_raw'] = hr_ov.apply(calculate_company_exposure, axis=1)
50
51 # Normalize คะแนนให้อยู่ในช่วง 0-10
52 def normalize_score(score, min_score, max_score):
53     """Normalize score to a scale of 0-10."""
54     if score == 'unknown':
55         return 'unknown'
56     normalized = (score - min_score) / (max_score - min_score) * 10
57     return min(max(normalized, 0), 10) # Ensure score is within 0-10
58
59 # Filter out 'unknown' values before calculating min and max
60 # แก้ไข: กรองค่า 'unknown' ออกก่อนคำนวณค่าต่ำสุดและสูงสุด
61 numerical_exposure = hr_ov['company_exposure_raw'][hr_ov['company_exposure_raw'] != 'unknown']
62 # แปลง numerical_exposure เป็นตัวเลข
63 numerical_exposure = pd.to_numeric(numerical_exposure)
64
65
66 # คำนวณค่าต่ำสุดและค่าสูงสุดของคะแนนจากข้อมูลที่เป็นตัวเลข
67 min_score = numerical_exposure.min()
68 max_score = numerical_exposure.max()
69
70 # Normalize คะแนน
71 hr_ov['company_exposure'] = hr_ov['company_exposure_raw'].apply(lambda x: normalize_score(x, min_score, max_score))
72
73 # ตรวจสอบค่าที่ได้ในคอลัมน์ company_exposure
74 print(hr_ov[['company_size', 'company_type', 'company_exposure_raw', 'company_exposure']].head())

```

	company_size	company_type	company_exposure_raw	company_exposure
0	unknown	unknown	unknown	unknown
1	50-99	Pvt Ltd	4.0	4.444444
2	unknown	unknown	unknown	unknown
3	unknown	Pvt Ltd	unknown	unknown
4	50-99	Funded Startup	4.8	5.925926



# Extract Data

- career\_progression

```
import pandas as pd

# Function to convert last_new_job to a score
def last_new_job_score(last_new_job):
    if last_new_job == '<1':
        return 1
    elif last_new_job == '1':
        return 2
    elif last_new_job == '2':
        return 3
    elif last_new_job == '3':
        return 4
    elif last_new_job == '4':
        return 5
    elif last_new_job == 'never':
        return 0
    else:
        return 6 # for '5+', or any unknown value

# Function to calculate career progression
def calculate_career_progression(row):
    # Get the last_new_job_score
    job_score = last_new_job_score(row['last_new_job'])

    # Calculate the career progression score
    career_progression = (row['education_score'] + row['experience_score']) / (1 + job_score)

    return career_progression
```

```
# Apply the function to calculate raw career progression scores
hr_ov['career_progression_raw'] = hr_ov.apply(calculate_career_progression, axis=1)

# Normalize career progression to a 0-10 scale
min_score = hr_ov['career_progression_raw'].min()
max_score = hr_ov['career_progression_raw'].max()

# Avoid division by zero
if max_score > min_score:
    hr_ov['career_progression'] = (hr_ov['career_progression_raw'] - min_score) / (max_score - min_score) * 10
else:
    hr_ov['career_progression'] = 0 # All values are the same if max == min

# Display the first few rows to verify
print(hr_ov[['education_score', 'experience_score', 'last_new_job', 'career_progression']].head())
```

	education_score	experience_score	last_new_job	career_progression
0	6.491228	10.0	1	2.649985
1	6.491228	10.0	>4	1.058046
2	8.245614	7.5	never	7.843897
3	5.789474	1.0	never	3.304966
4	8.245614	10.0	4	1.405225

# Extract Data

```
hr_ov = hr_ov.drop(['company_exposure_raw', 'career_progression_raw'], axis=1)
```

```
hr_ov.head(30)
```

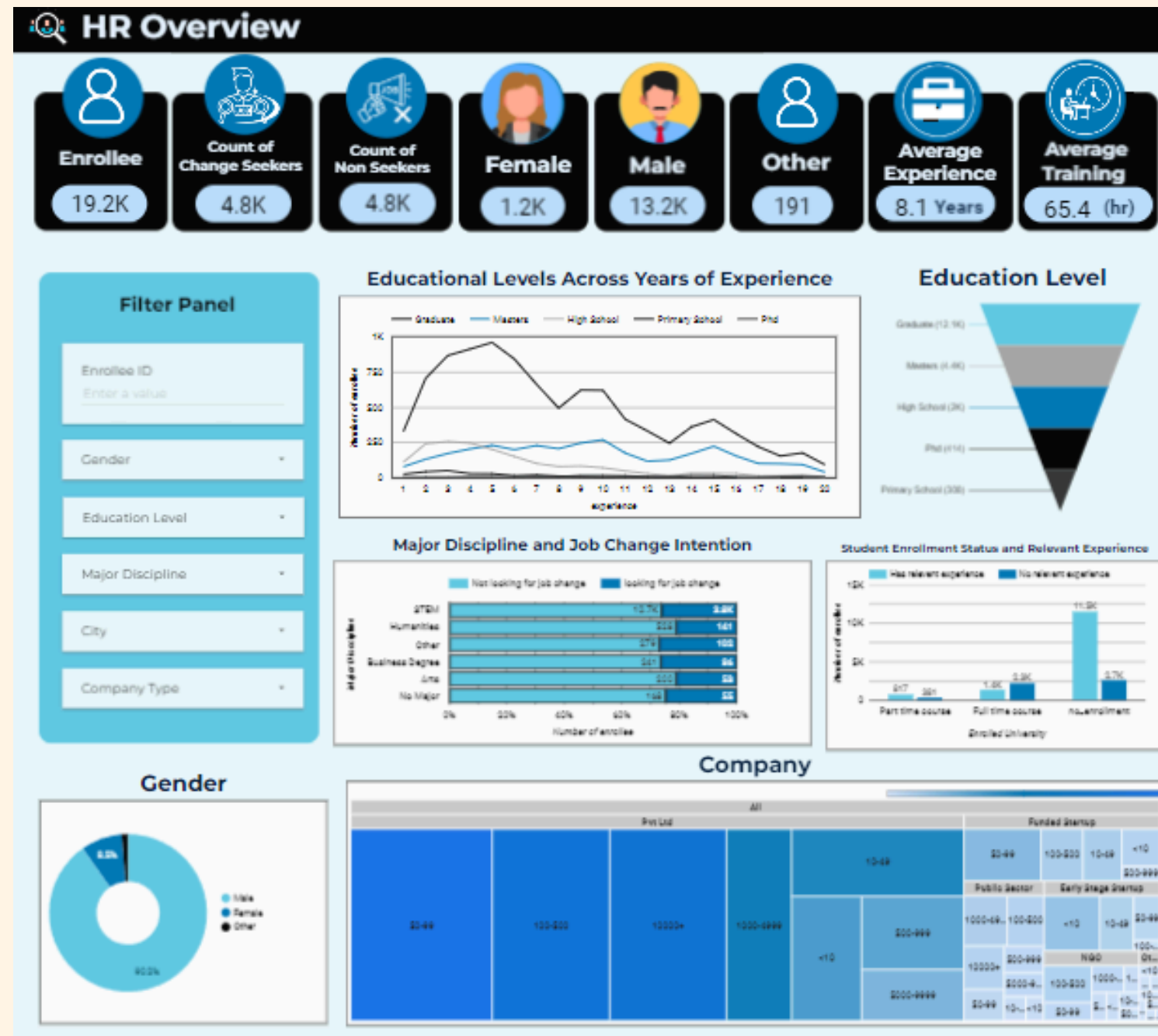
	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_discipline	experience	company_size	company_type	last_new_job	training_hours	target	education_score	experience_score	training_score	company_exposure	career_progression
0	8949	103	0.920	Male	Yes	no_enrollment	Graduate	STEM	>20	unknown	unknown	1	36	1.0	6.491228	10.0	1.071429	unknown	2.649988
1	29725	40	0.776	Male	No	no_enrollment	Graduate	STEM	15	50-99	Pvt Ltd	>4	47	0.0	6.491228	10.0	1.398810	4.444444	1.058046
2	11561	21	0.624	unknown	No	Full time course	Graduate	STEM	5	unknown	unknown	never	83	0.0	8.245614	7.5	2.470238	unknown	7.843897
3	33241	115	0.789	unknown	No	no_enrollment	Graduate	Business Degree	<1	unknown	Pvt Ltd	never	52	1.0	5.789474	1.0	1.547619	unknown	3.304966
4	666	162	0.767	Male	Yes	no_enrollment	Masters	STEM	>20	50-99	Funded Startup	4	8	0.0	8.245614	10.0	0.238095	5.925926	1.405223
5	21651	176	0.764	unknown	Yes	Part time course	Graduate	STEM	11	unknown	unknown	1	24	1.0	7.660819	10.0	0.714286	unknown	2.847566
6	28806	160	0.920	Male	Yes	no_enrollment	High School	unknown	5	50-99	Funded Startup	1	24	0.0	2.631579	9.5	0.714286	5.925926	1.913501

```
hr_ov.describe()
```

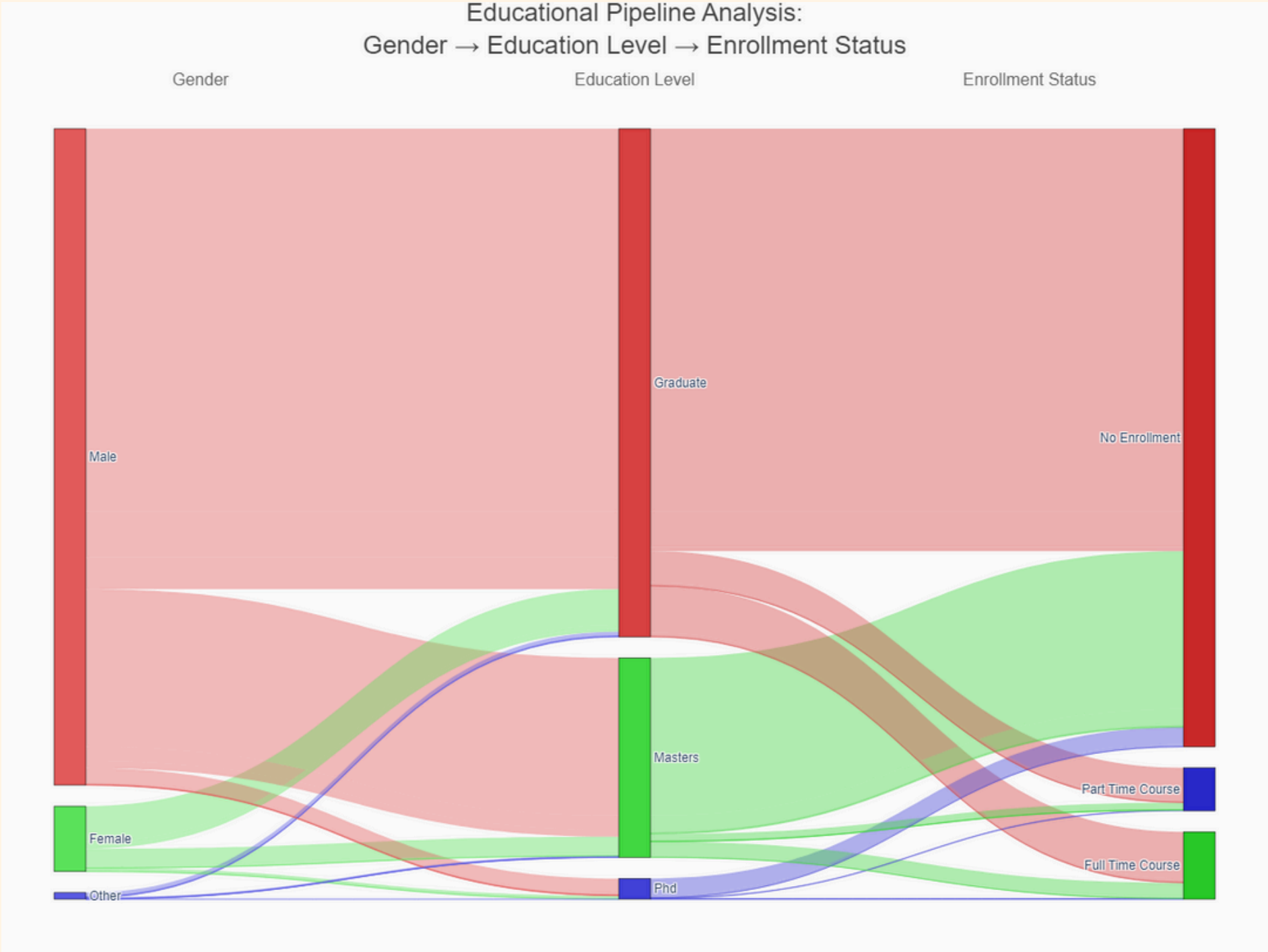
	enrollee_id	city	city_development_index	training_hours	target	education_score	experience_score	training_score	career_progression
count	19158.000000	19158.000000	19158.000000	19158.000000	19158.000000	19158.000000	19158.000000	19158.000000	19158.000000
mean	16875.358179	80.128876	0.828848	65.366896	0.249348	6.530813	8.510700	1.945443	2.418953
std	9616.292592	46.413570	0.123362	60.058462	0.432647	1.861612	2.553477	1.787454	1.693364
min	1.000000	1.000000	0.448000	1.000000	0.000000	0.701754	1.000000	0.029762	0.000000
25%	8554.250000	21.000000	0.740000	23.000000	0.000000	6.491228	8.000000	0.684524	1.273339
50%	16982.500000	101.000000	0.903000	47.000000	0.000000	6.491228	10.000000	1.398810	2.017232
75%	25169.750000	104.000000	0.920000	88.000000	0.000000	8.245614	10.000000	2.619048	2.649985
max	33380.000000	180.000000	0.949000	336.000000	1.000000	10.000000	10.000000	10.000000	10.000000

# Dashboard

page 10

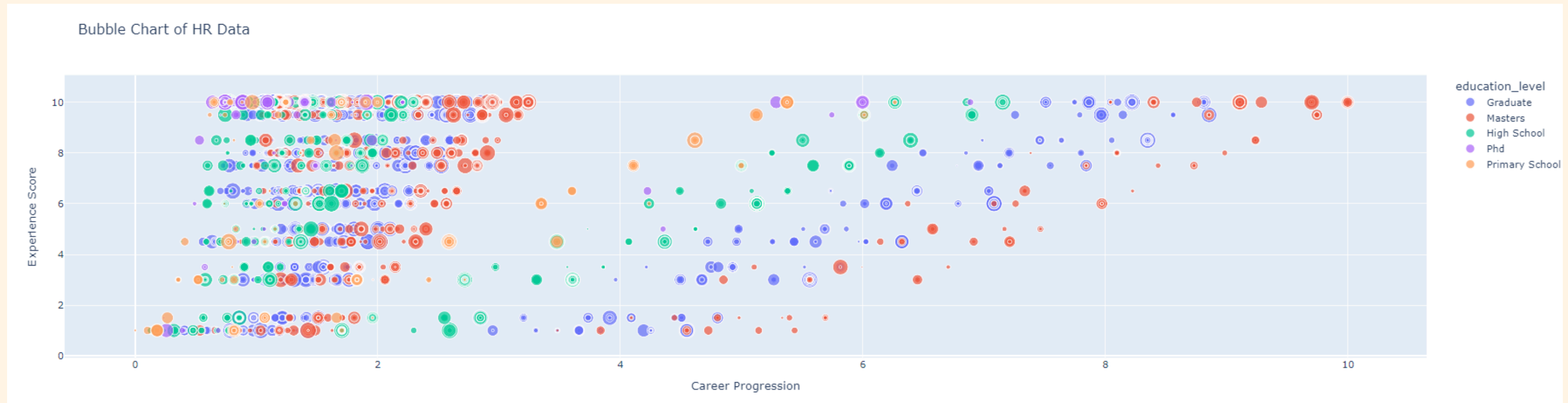


# Visualization





# Visualization





# Visualization

page 13

## HR Data Analysis: Parallel Coordinates Plot

☒ Graduate ☒ Masters ☒ High School ☒ Phd ☒ Primary School

Primary School  
Phd  
High School  
Master  
Graduate

