DS210 Final Project

Seo-Young (Emily) Yang

Collaborator: Solmin Lee

For my final project, I used the dataset, “Stress Detection” to find the relationship between the stress level and sleep quality. In the dataset, 100 participants were observed throughout 30 days and their stress level (PSS_score) and sleep quality (PSQI_score) was recorded daily. I wanted to find out if the stress level affects the sleep quality and vice versa.

To start off, I decided to find the average PSS_score and PSQI_score for each individual in one month period. After finding the average PSS_score and PSQI_score, I compared all 100 participants' average scores to find the correlation. To find these values I used the following rust code on my main.rs and data_loading.rs:

main.rs

```
        let dataset = Dataset::load(file_path)?;
        let averages = dataset.compute_averages();

        let x_values: Vec<f64> = averages.iter().map(|avg|
avg.pss).collect();
        let y_values: Vec<f64> = averages.iter().map(|avg|
avg.psqi).collect();
```

Data_loading.rs

```
use std::collections::HashMap;
use csv::StringRecord;

pub struct ParticipantAverage {
   pub pss: f64,
   pub psqi: f64,
```

```rust
}

pub struct Dataset {
    data: HashMap<u32, Vec<(f64, f64)>>,
}

impl Dataset {
    pub fn load(file_path: &str) -> Result<Self, Box<dyn
std::error::Error>> {
        let mut reader = csv::Reader::from_path(file_path)?;
        let mut data = HashMap::new();

        for result in reader.records() {
            let record: StringRecord = result?;
            let participant_id: u32 =
record.get(0).unwrap().parse()?;
            let pss: f64 = record.get(2).unwrap().parse()?;
            let psqi: f64 = record.get(11).unwrap().parse()?;

            data.entry(participant_id)
                .or_insert_with(Vec::new)
                .push((pss, psqi));
        }

        Ok(Self { data })
    }

    pub fn compute_averages(&self) -> Vec<ParticipantAverage> {
        self.data
            .iter()
            .map(|(_, scores)| {
                let (total_pss, total_psqi) =
scores.iter().fold((0.0, 0.0), |(acc_pss, acc_psqi), (pss, psqi)| {
```

```
                    (acc_pss + pss, acc_psqi + psqi)
                });
                let count = scores.len() as f64;
                ParticipantAverage {
                    pss: total_pss / count,
                    psqi: total_psqi / count,
                }
            })
            .collect()
    }
}
```

After finding the average PSS score and PSQI score of all 100 participants, I structured a rust code to find the correlation between two scores to see if the stress level impacts the sleep quality. To find the correlation I used the following codes:

main.rs

```
let correlation = pearson_correlation(&x_values, &y_values)?;
```

statistics.rs

```
ub fn pearson_correlation(x: &Vec<f64>, y: &Vec<f64>) -> Result<f64,
&'static str> {
    if x.len() != y.len() || x.is_empty() {
        return Err("Vectors must have the same non-zero length");
    }

    let n = x.len() as f64;
    let sum_x: f64 = x.iter().sum();
    let sum_y: f64 = y.iter().sum();
    let sum_xy: f64 = x.iter().zip(y.iter()).map(|(xi, yi)| xi *
yi).sum();
```

```
    let sum_x2: f64 = x.iter().map(|xi| xi * xi).sum();
    let sum_y2: f64 = y.iter().map(|yi| yi * yi).sum();

    let numerator = n * sum_xy - sum_x * sum_y;
    let denominator = ((n * sum_x2 - sum_x.powi(2)) * (n * sum_y2 -
sum_y.powi(2))).sqrt();

    Ok(numerator / denominator)
}
```

From this code, the correlation value was 0.027, which shows that there is a very weak correlation between the PSS score and PSQI score. Furthermore, I decided to find the linear regression model to find the predicted correlation between two variables. In order to find the linear regression model of the two variables, I used the following code:

main.rs

```
    let (slope, intercept) = linear_regression(&x_values, &y_values)?;
```

statistics.rs

```
pub fn linear_regression(x: &Vec<f64>, y: &Vec<f64>) -> Result<(f64,
f64), &'static str> {
    let n = x.len() as f64;
    let sum_x: f64 = x.iter().sum();
    let sum_y: f64 = y.iter().sum();
    let sum_xy: f64 = x.iter().zip(y.iter()).map(|(xi, yi)| xi *
yi).sum();
    let sum_x2: f64 = x.iter().map(|xi| xi * xi).sum();

    let slope = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 -
sum_x.powi(2));
    let intercept = (sum_y - slope * sum_x) / n;
```

```
    Ok((slope, intercept))
}
```

From this code, I was able to find the linear regression model which was Y = 2.395 + 0.004X. To better see the correlation between two variables, I decided to create the scatter plot of two variables and include the linear regression line so I can better visualize the relationship. To create the graph, I used to following code:

main.rs

```
let regression_file = "correlation_graph_with_regression.png";
    plot_correlation_graph(&x_values, &y_values, slope, intercept,
regression_file)?;
```

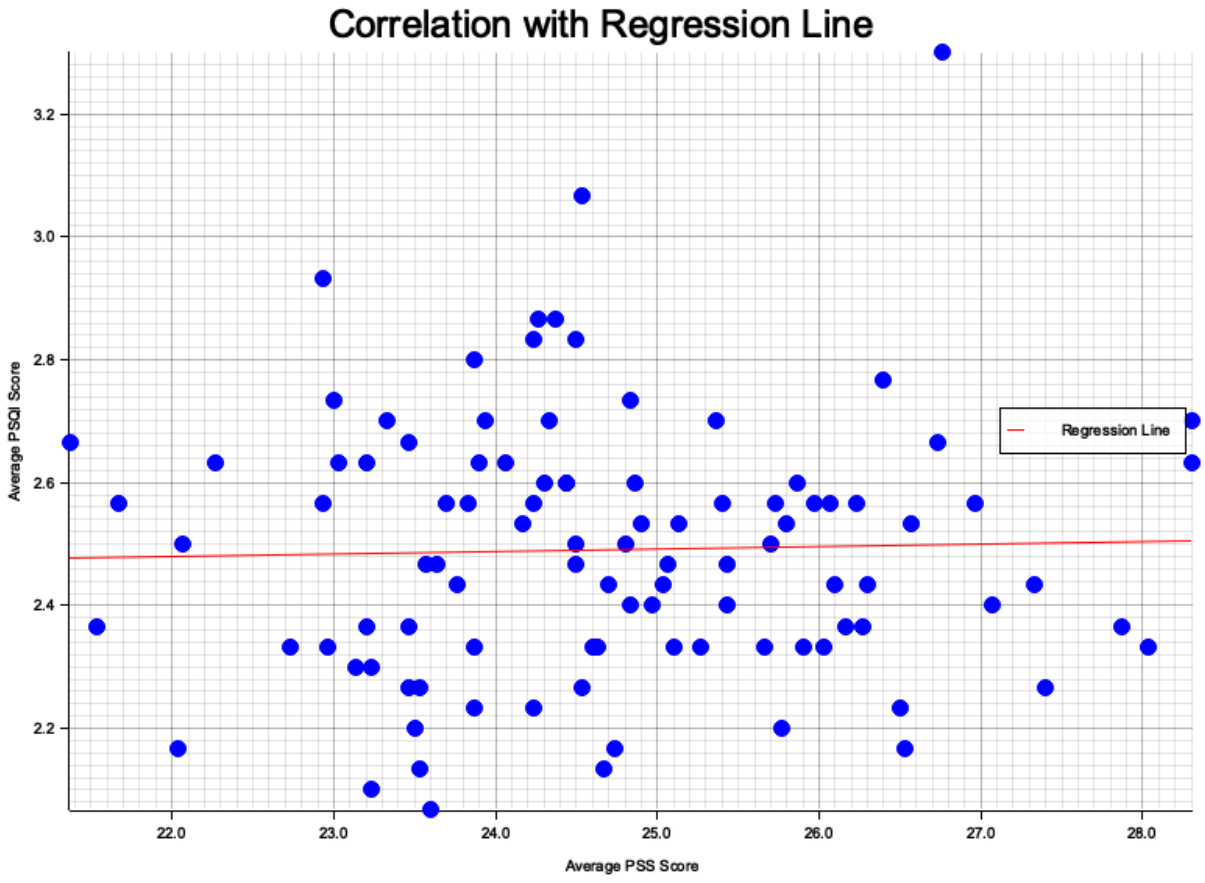
visualization.rs

```
pub fn plot_correlation_graph(
    x: &Vec<f64>,
    y: &Vec<f64>,
    slope: f64,
    intercept: f64,
    output_file: &str,
) -> Result<(), Box<dyn std::error::Error>> {
    let root = BitMapBackend::new(output_file, (800,
600)).into_drawing_area();
    root.fill(&WHITE)?;

    let x_min = *x.iter().min_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();
    let x_max = *x.iter().max_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();
```

```
    let y_min = *y.iter().min_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();
    let y_max = *y.iter().max_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();

    let mut chart = ChartBuilder::on(&root)
    .caption("Correlation with Regression Line", ("sans-serif", 30))
        .margin(20)
        .x_label_area_size(40)
        .y_label_area_size(40)
        .build_cartesian_2d(x_min..x_max, y_min..y_max)?;

    chart.configure_mesh()
        .x_desc("Average PSS Score")
        .y_desc("Average PSQI Score")
        .draw()?;

    chart.draw_series(
        x.iter()
            .zip(y.iter())
            .map(|(xi, yi)| Circle::new((*xi, *yi), 5,
BLUE.filled())),
    )?;
    let y_start = slope * x_min + intercept;
    let y_end = slope * x_max + intercept;

    chart.draw_series(LineSeries::new(
        vec![(x_min, y_start), (x_max, y_end)],
        &RED,
    ))?
    .label("Regression Line")
    .legend(|(x, y)| PathElement::new(vec![(x - 5, y), (x + 5, y)],
&RED));
```

```
    chart.configure_series_labels()
        .border_style(&BLACK)
        .background_style(&WHITE.mix(0.8))
        .draw()?;

    root.present()?;
    println!("Scatter plot with regression line saved to {}",
output_file);

    Ok(())
}
```

Through this code, I was able to find the visualization of the correlation and linear regression model of PSS score and PSQI score.



As we can see from the graph, the scatter plot shows that the relationship between the average PSS score and average PSQI has a very weak correlation which proves that our correlation is 0.027. Furthermore, the plots are not concentrated around the linear regression model which also proves that there is a weak correlation between two variables.

To better understand the correlation between PSS score and PSQI score using one more visualization, I decided to use heatmap. Through different colors, heatmap shows the correlation between two variables so we can quickly identify the correlation in a glance. I used the followings codes to find the heatmap:

main.rs

```
plot_heatmap(&x_values, &y_values, "heatmap_pss_psqi.png")?;
```

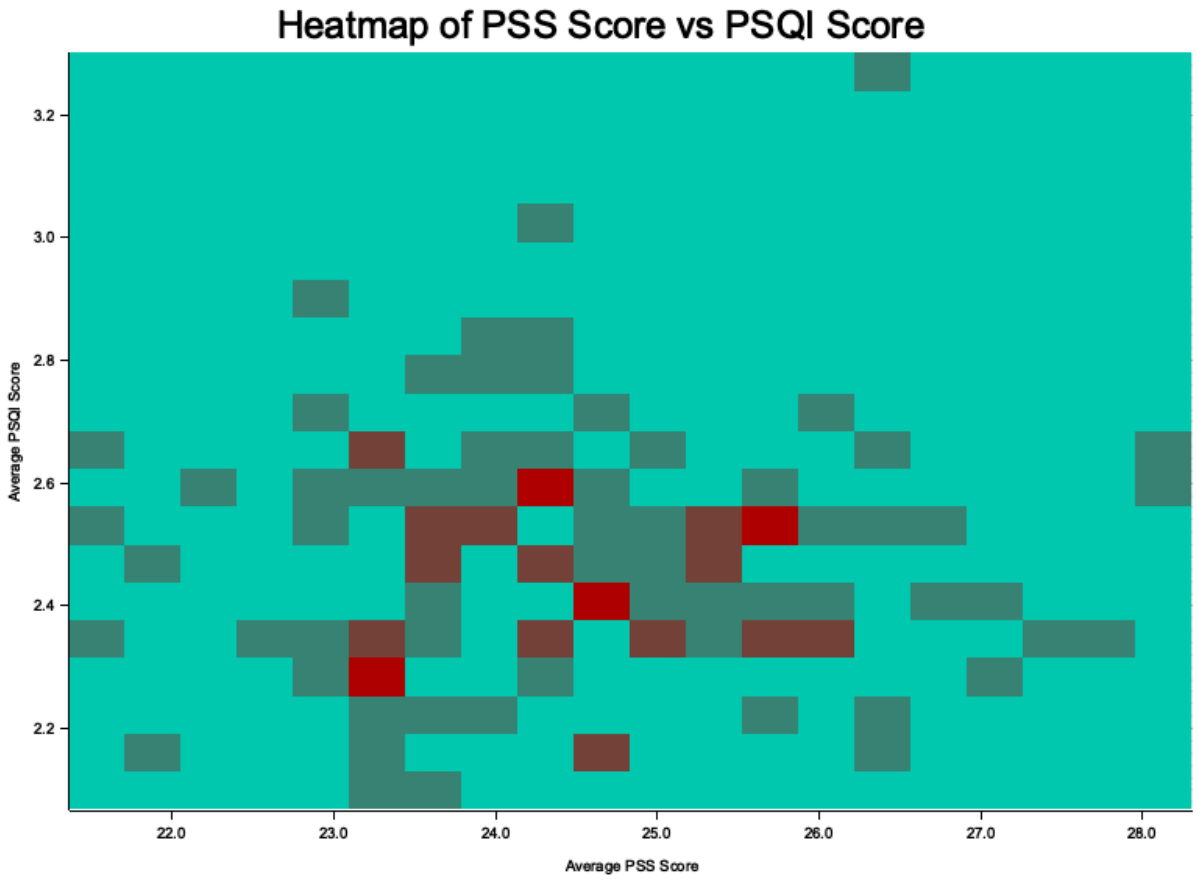
visualization.rs

```
pub fn plot_heatmap(
    x: &Vec<f64>,
    y: &Vec<f64>,
    output_file: &str,
) -> Result<(), Box<dyn std::error::Error>> {
    let root = BitMapBackend::new(output_file, (800,
600)).into_drawing_area();
    root.fill(&WHITE)?;

    let x_min = *x.iter().min_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();
    let x_max = *x.iter().max_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();
    let y_min = *y.iter().min_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();
```

```
    let y_max = *y.iter().max_by(|a, b|
a.partial_cmp(b).unwrap()).unwrap();

    let grid_size = 20;
    let mut grid = vec![vec![0; grid_size]; grid_size];

    for (&xi, &yi) in x.iter().zip(y.iter()) {
        let x_idx = ((xi - x_min) / (x_max - x_min) * (grid_size as
f64 - 1.0)) as usize;
        let y_idx = ((yi - y_min) / (y_max - y_min) * (grid_size as
f64 - 1.0)) as usize;
        grid[x_idx][y_idx] += 1;
    }

    let mut chart = ChartBuilder::on(&root)
        .caption("Heatmap of PSS Score vs PSQI Score", ("sans-serif",
30))
        .margin(20)
        .x_label_area_size(40)
        .y_label_area_size(40)
        .build_cartesian_2d(x_min..x_max, y_min..y_max)?;

    chart.configure_mesh()
        .x_desc("Average PSS Score")
        .y_desc("Average PSQI Score")
        .draw()?;

    let max_intensity =
grid.iter().flatten().copied().max().unwrap_or(1) as f64;

    for i in 0..grid_size {
        for j in 0..grid_size {
```

```
            let x0 = x_min + i as f64 / grid_size as f64 * (x_max -
x_min);
            let x1 = x_min + (i + 1) as f64 / grid_size as f64 *
(x_max - x_min);
            let y0 = y_min + j as f64 / grid_size as f64 * (y_max -
y_min);
            let y1 = y_min + (j + 1) as f64 / grid_size as f64 *
(y_max - y_min);

            let intensity = grid[i][j] as f64 / max_intensity;

            let red = (255.0 * intensity * 0.7) as u8;
            let green = (200.0 * (1.0 - intensity)) as u8;
            let blue = (255.0 * (1.0 - intensity) * 0.7) as u8;

            let color = RGBColor(red, green, blue);


            chart.draw_series(std::iter::once(Rectangle::new(
                [(x0, y0), (x1, y1)],
                color.filled(),
            )))?;
        }
    }

    root.present()?;
    Ok(())
}
```

After running this code, I was able to find the heatmap of the PSS score and PSQI score.



On this heatmap, the dark red defines the higher concentration and grey defines lower concentration and the teal color (basically the background color) indicates no data points. If we interpret this heatmap, we can see that participants in the dark red area have moderate correlation between PSS score and PSQI score. The gray area is more scattered throughout the heatmap which means that there are less participants who were included in these values. And the teal area means that there was no data that fell into those areas. We can see a clear concentration of data clustered around 24-26 PSS score and 2.4-2.6 PSQI score so we can define this as a common range of the participants. If we only consider the dark red data points, we can say that there is a positive correlation between two variables; however, we can't say that the correlation is strong.

To sum up, the dataset's results showed a positive relationship between the PSS scores and the PSQI scores. The heatmap highlights a clustering pattern showing that participants with higher stress levels can experience poorer sleep quality. However, the scatter plot and the linear regression model indicated that the correlation between the two variables is 0.027, which is a very weak correlation. This weak correlation proves that while a trend may exist, it is not strong enough to conclusively determine a direct relationship between stress levels and sleep quality within this dataset. Other factors or limitations may influence the outcome of the research, which is why further research is required to explore the relationship between stress level and sleeping quality using a larger and diverse dataset.