คู่มือการใช้งานโปรแกรม “🎨 สีคอลัมน์ R”

ต่อไปนี้คือคู่มือการใช้งานโปรแกรม “🎨 สีคอลัมน์ R” สำหรับชีต **Data** (คอลัมน์ **R**) ที่คุณติดตั้งไว้ครับ

คู่มือการใช้งาน “🎨 สีคอลัมน์ R”

1) โปรแกรมทำอะไรได้บ้าง

- ตั้ง **สีไล่ระดับ** ให้คอลัมน์ **R** ตามเกณฑ์ตัวเลข: `21–30`, `31–40`, `41–50`, `51–80`, `≥81`
- เลือก **โทนสี** ได้: 💜ม่วง / 💙ฟ้า / 💚เขียว / ❤️แดง / 💛เหลือง / 🧡ส้ม
- เลือกโหมดสีได้ 2 แบบ
    - **ช่วงคงที่ (Fixed ranges)**: ระบายสีตามช่วงตัวเลขตายตัว
    - **Color Scale**: ไล่สีอ่อน→เข้มแบบเนียนตามค่า (Min=21 / Mid=51 / Max=81)
- มีเมนู **สุ่มโทนสี** ทั้งแบบช่วงคงที่และแบบ Color Scale
- มีเมนู **ล้างกฎ**:
    - ล้างเฉพาะกฎที่ทับซ้อนคอลัมน์ R
    - ล้างเฉพาะกฎที่สคริปต์นี้สร้าง
    - ล้างกฎทั้งชีต Data

---

2) ขั้นตอนติดตั้ง (ครั้งแรก)

1. เปิด Google Sheet ที่ต้องการ (ต้องมีชีตชื่อ **Data**)
2. ไปที่ **Extensions** → **Apps Script**
3. วางโค้ด `code.gs` ที่ผมให้ลงไป (ทับของเดิมได้) → **Save**
4. ครั้งแรกให้กด **Run** ฟังก์ชันใดฟังก์ชันหนึ่งเพื่ออนุญาตสิทธิ์ (Authorize) ตามที่ระบบแจ้ง
5. กลับไปหน้า Spreadsheet แล้ว **รีเฟรชหน้าเว็บ**

6. จะเห็นเมนูใหม่บนแถบเมนูชื่อ “🎨 **สีคอลัมน์ R**”

ถ้าเมนูไม่ขึ้น ให้ลองรีเฟรชอีกครั้ง หรือรันฟังก์ชัน `onOpen()` ใน Apps Script หนึ่งครั้ง

---

3) ภาพรวมเมนูและคำสั่ง

เมนู: 🎨 **สีคอลัมน์ R**

- โทนสี (แต่ละโทนมี 2 คำสั่ง)
    - **ช่วงคงที่**: ตั้ง 5 ช่วงคงที่ 21–30, 31–40, 41–50, 51–80, ≥81
    - **Color Scale**: ไล่สีอัตโนมัติ Min=21, Mid=51, Max=81
- 🎲 **สุ่มโทนสี**
    - สุ่มโทน (ช่วงคงที่)
    - สุ่มโทน (Color Scale)
- 🧹 **ล้างกฎ**
    - ล้างกฎที่ทับซ้อนคอลัมน์ R
    - ล้างเฉพาะกฎที่สคริปต์นี้ตั้งในคอลัมน์ R
    - ล้างกฎทั้งชีต Data

---

4) วิธีใช้งานหลัก (Quick Start)

A) ตั้งสีโทนม่วงแบบช่วงคงที่

1. ไปที่เมนู 🎨 **สีคอลัมน์ R** → 💜 **ม่วง** → **ช่วงคงที่**
2. โปรแกรมจะลบกฎที่ทับซ้อนคอลัมน์ R และเพิ่มกฎใหม่ให้ทันที

B) ตั้งสีโทนม่วงแบบ Color Scale

1. ไปที่ 🎨 **สีคอลัมน์ R** → 💜 **ม่วง** → **Color Scale (21 → 81)**

2. โปรแกรมจะตั้งไล่สีอัตโนมัติจาก 21 (อ่อน) ถึง 81 (เข้ม)

C) สุ่มโทนสีแบบ Color Scale

1. ไปที่ 🎨 **สีคอลัมน์ R** → 🎲 **สุ่มโทนสี** → **สุ่มโทน (Color Scale)**
2. ระบบจะสุ่มเลือกโทนและตั้งไล่สีให้ทันที

D) ล้างกฎคอลัมน์ R

ไปที่ 🎨 **สีคอลัมน์ R** → 🧹 **ล้างกฎ** แล้วเลือกหนึ่งในตัวเลือก:

- **ล้างกฎที่ทับซ้อนคอลัมน์ R** (แนะนำ)
- **ล้างเฉพาะกฎที่สคริปต์นี้ตั้งในคอลัมน์ R**
- **ล้างกฎทั้งชีต Data** (แรงสุด รีเซ็ตทั้งหมด)

---

5) หลักการทำงานของสองโหมด

- **ช่วงคงที่ (Fixed ranges)**
  เหมาะเมื่อคุณต้องการ “สีตามเกณฑ์คะแนนที่กำหนดไว้ชัดเจน”
  - 21–30 = เฉดสีอ่อนสุด
  - 31–40
  - 41–50
  - 51–80
  - ≥81 = เฉดสีเข้มสุด
- **Color Scale**
  เหมาะเมื่อคุณต้องการไล่สีต่อเนื่องตามค่าแบบเนียน ๆ โดยตั้ง **Min=21, Mid=51, Max=81**

ทั้งสองโหมดจะปรับสีเฉพาะ **คอลัมน์ R ของชีต Data** เท่านั้น

---

6) เคล็ดลับการใช้งาน

- ถ้า **แถว R1** เป็นหัวตาราง ควรให้ข้อมูลเริ่มที่ **R2** เป็นต้นไป (ไม่จำเป็นต้องแก้โค้ด ระบบใช้สูตรล็อกคอลัมน์ `$R1` ซึ่งเลื่อนตามแถวอัตโนมัติได้)
- ถ้าข้อมูลบางเซลล์เป็น **ข้อความ (text)** สีจะไม่ทำงาน ให้แปลงเป็นตัวเลขจริง (เช่น เอา `,` ออก หรือใช้ `VALUE()` / Paste special → **Values only**)
- ถ้ามี **กฎเดิมหลายชุด** แนะนำใช้ **"ล้างกฎที่ทับซ้อนคอลัมน์ R"** ก่อนตั้งกฎใหม่ทุกครั้ง เพื่อลดการทับซ้อน
- ถ้าคอลัมน์ R ถูก **ป้องกัน (Protected range)** หรือชีตถูกล็อก ให้ปลดล็อกก่อนตั้งกฎ

---

7) การปรับแต่งที่พบบ่อย

7.1 เปลี่ยนช่วงทำงาน (เฉพาะบางแถว)

- แทน `R:R` เป็นช่วงที่ต้องการ เช่น `R2:R500`
    - แก้ในบรรทัดที่เรียกใช้ฟังก์ชันตั้งสี (ในฟังก์ชัน `applyFixedRules` / `applyColorScale` ที่เรียกด้วย `'R:R'`)

7.2 เปลี่ยนชื่อชีต

- แก้ `'Data'` เป็นชื่อชีตของคุณ ในฟังก์ชันเรียกใช้ (เช่น `applyFixedRules('ชื่อชีต','R:R',...)`)

7.3 เปลี่ยนเกณฑ์ตัวเลข (21, 31, 41, 51, 81)

โหมด "ช่วงคงที่": แก้สูตรใน `applyFixedRules()` ที่กำหนดเป็น

=AND(ISNUMBER($R1), $R1>=ค่าต่ำ, $R1<ค่าสูง)

-
- โหมด Color Scale: แก้ค่า Min/Mid/Max ใน `applyColorScale()` (ค่า '21' / '51' / '81')

7.4 เพิ่มโทนสีใหม่

เพิ่มรายการในคงที่ `PALETTES` เช่น

'💗 ชมพู': ['#FCE4EC','#F8BBD0','#F06292','#E91E63','#880E4F']

-

- แล้วสร้างฟังก์ชันเรียกใช้คล้าย ๆ โทนอื่น ๆ

---

8) ปัญหาที่พบบ่อย & วิธีแก้

Q: กด “ล้างกฎ” แล้วบางกฎยังอยู่?
A: ใช้ตัวเลือก **“ล้างกฎที่ทับซ้อนคอลัมน์ R”** ซึ่งจะลบทุกกฎที่ช่วง **แตะ** คอลัมน์ R (แม้กฎนั้นครอบคอลัมน์อื่นด้วย) หากยังมี ให้ใช้ **“ล้างกฎทั้งชีต Data”** (จะล้างหมดทั้งชีต)

Q: ทำไมสีไม่ขึ้น?

- ค่าใน R เป็น **ข้อความ** ไม่ใช่ตัวเลข → แปลงเป็นตัวเลข
- ค่าต่ำกว่า 21 (นอกช่วง) ในโหมดช่วงคงที่ → จะไม่ติดสี
- มี **กฎอื่น** ทับซ้อนและอยู่ลำดับบนกว่า → ล้างกฎก่อน หรือจัดลำดับใหม่

Q: เมนูไม่แสดงหลังวางโค้ด?

- บันทึกโค้ดแล้ว **รีเฟรช**
- รัน `onOpen()` ใน Apps Script หนึ่งครั้ง
- ตรวจสิทธิ์ (Authorize) ให้ครบ

Q: ใช้กับหลายชีตได้ไหม?

- ได้ โดยคัดลอกโค้ดไปใช้ และเปลี่ยนชื่อชีต/ช่วงตามต้องการ หรือเพิ่มเมนูย่อยให้เลือกชีต

---

9) โทนสีที่ใช้ (อ้างอิง)

- 💜 ม่วง: `#E1BEE7, #CE93D8, #BA68C8, #8E24AA, #6A1B9A`
- 💙 ฟ้า: `#BBDEFB, #90CAF9, #42A5F5, #1E88E5, #0D47A1`
- 💚 เขียว: `#C8E6C9, #A5D6A7, #66BB6A, #388E3C, #1B5E20`
- ❤️ แดง: `#FFCDD2, #EF9A9A, #E57373, #E53935, #B71C1C`

- 💛 เหลือง: `#FFF9C4, #FFF59D, #FFF176, #FBC02D, #F57F17`
- 🧡 ส้ม: `#FFE0B2, #FFCC80, #FFB74D, #FB8C00, #E65100`

---

10) ตัวอย่างการใช้งานด่วน (แนะนำ)

1. ต้องการโทนม่วงแบบช่วงคงที่ → 🎨 **สีคอลัมน์ R** → 💜 **ม่วง** → **ช่วงคงที่**
2. เปลี่ยนใจอยากลองสีเขียวแบบ Color Scale → 💚 **เขียว** → **Color Scale**
3. อยากรีเซ็ตคอลัมน์ R → 🧹 **ล้างกฎ** → **ล้างกฎที่ทับซ้อนคอลัมน์ R**
4. ลองสุ่มโทนสวย ๆ → 🎲 **สุ่มโทนสี** → **สุ่มโทน (Color Scale)**

หากต้องการ “จำโทนล่าสุด” เพื่อเรียกใช้ซ้ำจากเมนูด่วน หรืออยากจำกัดช่วงเฉพาะ `R2:R500` เป็นค่าเริ่มต้น บอกผมได้เลย เดี๋ยวผมปรับโค้ดให้พร้อมใช้ให้ครับ 👍

formatColor.gs

```
/**********************
 * 🎨 R Column Styler
 * Sheet: 'Data', Column: R
 * Ranges: 21–30, 31–40, 41–50, 51–80, ≥81
 **********************/

function onOpen() {
  const ui = SpreadsheetApp.getUi();
  ui.createMenu('🎨 สีคอลัมน์ R')
    .addSubMenu(
      ui.createMenu('💜 ม่วง')
        .addItem('ช่วงคงที่ (21–30,31–40,41–50,51–80,≥81)', 'setupRangingPurple')
        .addItem('Color Scale (21→81)', 'setupColorScalePurple')
    )
    .addSubMenu(
      ui.createMenu('💙 ฟ้า')
        .addItem('ช่วงคงที่', 'setupRangingBlue')
        .addItem('Color Scale', 'setupColorScaleBlue')
    )
    .addSubMenu(
      ui.createMenu('💚 เขียว')
        .addItem('ช่วงคงที่', 'setupRangingGreen')
        .addItem('Color Scale', 'setupColorScaleGreen')
    )
    .addSubMenu(
      ui.createMenu('❤️ แดง')
        .addItem('ช่วงคงที่', 'setupRangingRed')
        .addItem('Color Scale', 'setupColorScaleRed')
    )
    .addSubMenu(
      ui.createMenu('💛 เหลือง')
        .addItem('ช่วงคงที่', 'setupRangingYellow')
        .addItem('Color Scale', 'setupColorScaleYellow')
    )
    .addSubMenu(
      ui.createMenu('🧡 ส้ม')
        .addItem('ช่วงคงที่', 'setupRangingOrange')
        .addItem('Color Scale', 'setupColorScaleOrange')
    )
    .addSubMenu(
      ui.createMenu('🎲 สุ่มโทนสี')
        .addItem('สุ่มโทน (ช่วงคงที่)', 'setupRangingRandom')
        .addItem('สุ่มโทน (Color Scale)', 'setupColorScaleRandom')
    )
    .addSubMenu(
      ui.createMenu('🧹 ล้างกฎ')
        .addItem('ล้างกฎที่ทับซ้อนคอลัมน์ R', 'clearRulesR')
        .addItem('ล้างเฉพาะกฎที่สคริปต์นี้ตั้งในคอลัมน์ R', 'clearOnlyOurRulesInR')
```

```
      .addItem('ล้างกฎทั้งชีต Data', 'clearAllRulesInSheet')
  )
  .addToUi();
}

/* ========= พาเลตสี (Material-ish) ========= */
const PALETTES = {
  '💜 ม่วง' : ['#E1BEE7','#CE93D8','#BA68C8','#8E24AA','#6A1B9A'],
  '💙 ฟ้า'  : ['#BBDEFB','#90CAF9','#42A5F5','#1E88E5','#0D47A1'],
  '💚 เขียว': ['#C8E6C9','#A5D6A7','#66BB6A','#388E3C','#1B5E20'],
  '❤️ แดง'  : ['#FFCDD2','#EF9A9A','#E57373','#E53935','#B71C1C'],
  '💛 เหลือง': ['#FFF9C4','#FFF59D','#FFF176','#FBC02D','#F57F17'],
  '🧡 ส้ม'  : ['#FFE0B2','#FFCC80','#FFB74D','#FB8C00','#E65100']
};

/* ========= ฟังก์ชันกลาง: ตั้งช่วงคงที่ / Color scale ========= */
function applyFixedRules(sheetName, rangeA1, colors, label) {
  const sh = SpreadsheetApp.getActive().getSheetByName(sheetName);
  if (!sh) throw new Error('ไม่พบชีตชื่อ ' + sheetName);
  const range = sh.getRange(rangeA1);

  // ล้างกฎที่ "ทับซ้อน" กับคอลัมน์เป้าหมายก่อน
  clearConditionalInRange_(sh, range);

  const rules = sh.getConditionalFormatRules();
  const makeRule = (formula, bg) =>
    SpreadsheetApp.newConditionalFormatRule()
      .whenFormulaSatisfied(formula)
      .setBackground(bg)
      .setFontColor('#000000')
      .setRanges([range])
      .build();

  // ช่วงคงที่ 21–30, 31–40, 41–50, 51–80, ≥81 (ล็อกคอลัมน์ R แต่ให้เลื่อนแถวได้)
  const r1 = makeRule('=AND(ISNUMBER($R1), $R1>=21, $R1<31)', colors[0]);
  const r2 = makeRule('=AND(ISNUMBER($R1), $R1>=31, $R1<41)', colors[1]);
  const r3 = makeRule('=AND(ISNUMBER($R1), $R1>=41, $R1<51)', colors[2]);
  const r4 = makeRule('=AND(ISNUMBER($R1), $R1>=51, $R1<81)', colors[3]);
  const r5 = makeRule('=AND(ISNUMBER($R1), $R1>=81)',          colors[4]);

  rules.push(r1, r2, r3, r4, r5);
  sh.setConditionalFormatRules(rules);
  SpreadsheetApp.getActive().toast('ตั้งค่าสำเร็จ: ' + label + ' (ช่วงคงที่)', '🎨 สีคอลัมน์ R', 4);
}

function applyColorScale(sheetName, rangeA1, minColor, midColor, maxColor, label) {
  const sh = SpreadsheetApp.getActive().getSheetByName(sheetName);
  if (!sh) throw new Error('ไม่พบชีตชื่อ ' + sheetName);
```

```
  const range = sh.getRange(rangeA1);

  clearConditionalInRange_(sh, range);

  const rules = sh.getConditionalFormatRules();
  const rule = SpreadsheetApp.newConditionalFormatRule()
    .setGradientMinpointWithValue(SpreadsheetApp.InterpolationType.NUMBER, '21',
minColor)
    .setGradientMidpointWithValue(SpreadsheetApp.InterpolationType.NUMBER, '51',
midColor)
    .setGradientMaxpointWithValue(SpreadsheetApp.InterpolationType.NUMBER, '81',
maxColor)
    .setRanges([range])
    .build();

  rules.push(rule);
  sh.setConditionalFormatRules(rules);
  SpreadsheetApp.getActive().toast('ตั้งค่าสำเร็จ: ' + label + ' (Color Scale)', '🎨 สีคอลัมน์ R',
4);
}

/* ========= ยูทิลิตี้: ตรวจทับซ้อนและล้างกฎ ========= */
// ตรวจว่าช่วงสองช่วงทับซ้อนกันหรือไม่
function rangesIntersect_(aRange, bRange) {
  const aRow1 = aRange.getRow(), aRow2 = aRow1 + aRange.getNumRows() - 1;
  const aCol1 = aRange.getColumn(), aCol2 = aCol1 + aRange.getNumColumns() - 1;
  const bRow1 = bRange.getRow(), bRow2 = bRow1 + bRange.getNumRows() - 1;
  const bCol1 = bRange.getColumn(), bCol2 = bCol1 + bRange.getNumColumns() - 1;
  const rowsOverlap = !(aRow2 < bRow1 || bRow2 < aRow1);
  const colsOverlap = !(aCol2 < bCol1 || bCol2 < aCol1);
  return rowsOverlap && colsOverlap;
}

// ล้างกฎทุกอันที่ “ทับซ้อน” กับช่วงที่ให้มา (ใช้ได้จริงกับกรณี R:R)
function clearConditionalInRange_(sheet, targetRange) {
  const oldRules = sheet.getConditionalFormatRules();
  const keep = [];
  for (const rule of oldRules) {
    const touches = rule.getRanges().some(rg => rangesIntersect_(rg, targetRange));
    if (!touches) keep.push(rule);
  }
  sheet.setConditionalFormatRules(keep);
  return oldRules.length - keep.length;
}

// ล้างกฎทั้งชีต
function clearAllRulesInSheet() {
  const sh = SpreadsheetApp.getActive().getSheetByName('Data');
```

```
  if (!sh) throw new Error('ไม่พบชีตชื่อ Data');
  sh.setConditionalFormatRules([]);
  SpreadsheetApp.getActive().toast('ล้างกฎ Conditional Formatting ทั้งชีตแล้ว', '🧹 เคลียร์กฎ',
4);
}

// ล้างเฉพาะ “กฎที่สคริปต์นี้ตั้ง” ในคอลัมน์ R (จับจากสูตร/ค่าสี/ช่วง)
function clearOnlyOurRulesInR() {
  const sh = SpreadsheetApp.getActive().getSheetByName('Data');
  if (!sh) throw new Error('ไม่พบชีตชื่อ Data');
  const target = sh.getRange('R:R');
  const oldRules = sh.getConditionalFormatRules();
  const OUR_FORMULAS = new Set([
    '=AND(ISNUMBER($R1), $R1>=21, $R1<31)',
    '=AND(ISNUMBER($R1), $R1>=31, $R1<41)',
    '=AND(ISNUMBER($R1), $R1>=41, $R1<51)',
    '=AND(ISNUMBER($R1), $R1>=51, $R1<81)',
    '=AND(ISNUMBER($R1), $R1>=81)'
  ]);

  const keep = [];
  let removed = 0;

  for (const rule of oldRules) {
    const touchesR = rule.getRanges().some(rg => rangesIntersect_(rg, target));
    let isOurs = false;

    // จับกฎแบบสูตรคงที่
    const cond = rule.getBooleanCondition && rule.getBooleanCondition();
    if (cond) {
      const type = cond.getCriteriaType && cond.getCriteriaType();
      const args = cond.getCriteriaValues && cond.getCriteriaValues();
      // whenFormulaSatisfied เก็บสูตรไว้ใน args[0]
      if (type === SpreadsheetApp.BooleanCriteria.CUSTOM_FORMULA && args &&
OUR_FORMULAS.has(String(args[0]).trim())) {
        isOurs = true;
      }
    }

    // จับกฎแบบไล่สีที่มีค่าคงที่ 21 / 51 / 81 (ตามที่สคริปต์ตั้ง)
    const grad = rule.getGradientCondition && rule.getGradientCondition();
    if (!isOurs && grad) {
      const min = grad.getMinpoint() || {};
      const mid = grad.getMidpoint() || {};
      const max = grad.getMaxpoint() || {};
      // เงื่อนไขของเรา: NUMBER=21, NUMBER=51, NUMBER=81
      const looksLikeOurs =
```

```
       min.type === SpreadsheetApp.InterpolationType.NUMBER && String(min.value) ===
'21' &&
       mid.type === SpreadsheetApp.InterpolationType.NUMBER && String(mid.value) ===
'51' &&
       max.type === SpreadsheetApp.InterpolationType.NUMBER && String(max.value) ===
'81';
      if (looksLikeOurs) isOurs = true;
    }

    if (touchesR && isOurs) {
      removed++;
      continue; // drop
    }
    keep.push(rule);
  }

  sh.setConditionalFormatRules(keep);
  SpreadsheetApp.getActive().toast(`ลบกฎของสคริปต์นี้ในคอลัมน์ R แล้ว ${removed} รายการ`,
'🧹 เคลียร์กฎ', 4);
}

/* ========= เมนู: ล้างกฎที่ทับซ้อนคอลัมน์ R ========= */
function clearRulesR() {
  const sh = SpreadsheetApp.getActive().getSheetByName('Data');
  if (!sh) throw new Error('ไม่พบชีตชื่อ Data');
  const target = sh.getRange('R:R');
  const removed = clearConditionalInRange_(sh, target);
  SpreadsheetApp.getActive().toast(`ลบกฎที่ทับซ้อนคอลัมน์ R แล้ว ${removed} รายการ`, '🧹
เคลียร์กฎ', 4);
}

/* ========= ตัวเรียกใช้งานสำหรับแต่ละโทน ========= */
function setupRangingPurple(){ applyFixedRules('Data','R:R', PALETTES['💜 ม่วง'], '💜 ม่วง');
}
function setupColorScalePurple(){ applyColorScale('Data','R:R', PALETTES['💜 ม่วง'][0],
PALETTES['💜 ม่วง'][2], PALETTES['💜 ม่วง'][4], '💜 ม่วง'); }

function setupRangingBlue(){ applyFixedRules('Data','R:R', PALETTES['💙 ฟ้า'], '💙 ฟ้า'); }
function setupColorScaleBlue(){ applyColorScale('Data','R:R', PALETTES['💙 ฟ้า'][0],
PALETTES['💙 ฟ้า'][2], PALETTES['💙 ฟ้า'][4], '💙 ฟ้า'); }

function setupRangingGreen(){ applyFixedRules('Data','R:R', PALETTES['💚 เขียว'], '💚 เขียว
'); }
function setupColorScaleGreen(){ applyColorScale('Data','R:R', PALETTES['💚 เขียว'][0],
PALETTES['💚 เขียว'][2], PALETTES['💚 เขียว'][4], '💚 เขียว'); }

function setupRangingRed(){ applyFixedRules('Data','R:R', PALETTES['❤️ แดง'], '❤️ แดง'); }
```

```
function setupColorScaleRed(){ applyColorScale('Data','R:R', PALETTES['❤️ แดง'][0], PALETTES['❤️ แดง'][2], PALETTES['❤️ แดง'][4], '❤️ แดง'); }

function setupRangingYellow(){ applyFixedRules('Data','R:R', PALETTES['💛 เหลือง'], '💛 เหลือง'); }
function setupColorScaleYellow(){ applyColorScale('Data','R:R', PALETTES['💛 เหลือง'][0], PALETTES['💛 เหลือง'][2], PALETTES['💛 เหลือง'][4], '💛 เหลือง'); }

function setupRangingOrange(){ applyFixedRules('Data','R:R', PALETTES['🧡 ส้ม'], '🧡 ส้ม'); }
function setupColorScaleOrange(){ applyColorScale('Data','R:R', PALETTES['🧡 ส้ม'][0], PALETTES['🧡 ส้ม'][2], PALETTES['🧡 ส้ม'][4], '🧡 ส้ม'); }

/* ========= สุ่มโทน ========= */
function setupRangingRandom() {
  const keys = Object.keys(PALETTES);
  const pick = keys[Math.floor(Math.random() * keys.length)];
  applyFixedRules('Data','R:R', PALETTES[pick], pick + ' • Random');
}
function setupColorScaleRandom() {
  const keys = Object.keys(PALETTES);
  const pick = keys[Math.floor(Math.random() * keys.length)];
  const p = PALETTES[pick];
  applyColorScale('Data','R:R', p[0], p[2], p[4], pick + ' • Random');
}
```

แนวคิด (Concept)

ได้เลยครับ — นี่คือ “ขั้นตอนการสร้างสูตร” + “แนวคิด” แบบจับมือทำ สำหรับการไล่สีคอลัมน์ **R** ในชีต **Data** ตามช่วง `21–30, 31–40, 41–50, 51–80, ≥81` ด้วยสีม่วงจากอ่อนไปเข้ม

แนวคิด (Concept)

1. **กำหนดช่วงแบบไม่ซ้อนทับ (Non-overlapping)**
   ใช้หลัก “**ปิดที่ขอบล่าง เปิดที่ขอบบน**” คือ `>=` ค่าต่ำ และ `<` ค่าสูง ยกเว้นช่วงสุดท้าย `>=81` เพื่อกันชนกันพอดีที่ 31, 41, 51, 81
   - 21–30: `>=21` และ `<31`
   - 31–40: `>=31` และ `<41`
   - 41–50: `>=41` และ `<51`
   - 51–80: `>=51` และ `<81`
   - ≥81: `>=81`
2. **ล็อกคอลัมน์ แต่ให้เลื่อนตามแถว**
   ใน Conditional formatting สูตรจะอ้างอิง “แถวแรกของช่วงที่เลือก” เป็นจุดอ้างอิง
   - ถ้าเลือกทั้งคอลัมน์ `R:R` → แถวแรกคือ `R1` ใช้ `$R1`
   - ถ้าเลือกตั้งแต่ `R2:R` → แถวแรกคือ `R2` ใช้ `$R2`
     หลักคือ **ตัวเลขหลัง R ต้องตรงกับ “แถวบนสุดของช่วงที่คุณเลือก”** เสมอ
3. **กันข้อมูลไม่ใช่ตัวเลข**
   ใช้ `ISNUMBER()` ครอบ เพื่อไม่ให้ช่องว่าง/ข้อความติดสีผิดพลาด
4. **สี: อ่อนไปเข้ม (โทนม่วง Material-ish)**
   - 21–30 → `#E1BEE7`
   - 31–40 → `#CE93D8`
   - 41–50 → `#BA68C8`
   - 51–80 → `#8E24AA`

- ≥81 → #6A1B9A

5. **สองแนวทาง**
   - **Fixed ranges (หลายกฎ)**: คุมได้เป๊ะตามเกณฑ์
   - **Color scale (กฎเดียว)**: ไล่สีเนียนตามค่า (ตั้ง Min=21, Mid=51, Max=81)

ขั้นตอนสร้าง “สูตร” แบบ Fixed ranges (Manual)

ตัวอย่างนี้สมมติคุณเลือก **ทั้งคอลัมน์ R (R:R)** ถ้าเลือก R2:R ให้เปลี่ยน $R1 เป็น $R2

1. ไฮไลต์ช่วงที่จะลงสี → **R:R**
2. ไปที่ **Format → Conditional formatting → Add another rule**
3. เลือก **Custom formula is** แล้วใส่สูตรตามช่วงด้านล่าง (สร้าง 5 กฎ)

- ช่วง **21–30** (ม่วงอ่อน #E1BEE7)

```
=AND(ISNUMBER($R1), $R1>=21, $R1<31)
```

- ช่วง **31–40** (ม่วงอ่อนกลาง #CE93D8)

```
=AND(ISNUMBER($R1), $R1>=31, $R1<41)
```

- ช่วง **41–50** (ม่วงกลาง #BA68C8)

```
=AND(ISNUMBER($R1), $R1>=41, $R1<51)
```

- ช่วง **51–80** (ม่วงเข้ม #8E24AA)

=AND(ISNUMBER($R1), $R1>=51, $R1<81)

- ช่วง **≥81** (ม่วงเข้มมาก #6A1B9A)

=AND(ISNUMBER($R1), $R1>=81)

4. ตั้ง **Fill color** ของแต่ละกฎตามโค้ดสีด้านบน
5. เรียงลำดับกฎไม่จำเป็นมากนัก (เพราะช่วงไม่ทับกัน) แต่เพื่อความสบายใจจะเรียงจากช่วงล่าง →บนก็ได้
6. กด **Done**

ถ้าเลือกช่วงเริ่มที่ **R2:R** ให้แทน $R1 ทั้งหมดเป็น $R2 (ย้ำ: ตัวเลขต้องเท่ากับแถวบนสุดของช่วงที่คุณเลือก)

ขั้นตอนแบบ Color scale (ไล่สีเนียนกฎเดียว)

1. ไฮไลต์ช่วง **R:R**
2. **Format → Conditional formatting → แท็บ Color scale**
3. ตั้งค่า:
    - **Min** = *Number* → **21** → สี #E1BEE7
    - **Midpoint** = *Number* → **51** → สี #BA68C8
    - **Max** = *Number* → **81** → สี #6A1B9A
4. กด **Done**

เหมาะเมื่อคะแนน/ค่ากระจายต่อเนื่อง ต้องการไล่เฉดอ่อน→เข้มแบบ Smooth

---

เช็คลิสต์ทดสอบ (Debug Checklist)

- ช่องว่าง/ข้อความไม่ควรติดสี → ถ้าติด ให้ตรวจว่าใช้ `ISNUMBER()` แล้ว
- ค่า **30** ต้องอยู่ช่วงแรก (21–30) เพราะสูตรใช้ `<31`
- ค่า **31** ต้องติดกฎช่วง 31–40 (เพราะ `>=31`)
- เลือกช่วง **ถูกจุดเริ่ม** หรือไม่ (ถ้าเลือก R2:R แต่ยังใช้ `$R1` จะอ้างอิงแถวผิด)
- มี **กฎอื่นทับอยู่** หรือเปล่า → ลองล้างกฎซ้ำซ้อนก่อน (ใช้เมนูที่ผมให้ก็ได้)

ทางเลือก/ทริคเพิ่ม

แบบไม่เขียนหลายกฎ (ช่วยตรวจช่วงในคอลัมน์ช่วย)
ใส่สูตรช่วยในคอลัมน์เสริม (เช่น S) เพื่อบอกช่วง แล้วกำหนดสีตามข้อความได้

```
=IF( NOT(ISNUMBER(R2)), "",
  IF(R2<21, "ต่ำกว่า 21",
  IF(R2<31, "21–30",
  IF(R2<41, "31–40",
  IF(R2<51, "41–50",
  IF(R2<81, "51–80", "81+"))))))
```

- จากนั้นตั้ง CF แบบ “Text is exactly” ต่อ 5 ช่วง (เข้าใจง่ายตอนแก้เกณฑ์)
- **ต้องครอบทั้งชีตอย่างเร็ว** → ใช้ **Apps Script** ที่ผมให้ไว้ จะเซ็ตทุกกฎให้คอลัมน์ R แบบคลิกครั้งเดียว

ถ้าต้องการสรุป **ไฟล์สั้น ๆ (PDF/ภาพ)** แผ่นเดียวที่มี “สูตรทุกช่วง + ขั้นตอน + สีที่ใช้” บอกได้เลย เดี๋ยวผมแพ็กให้พร้อมใช้งานครับ 💜