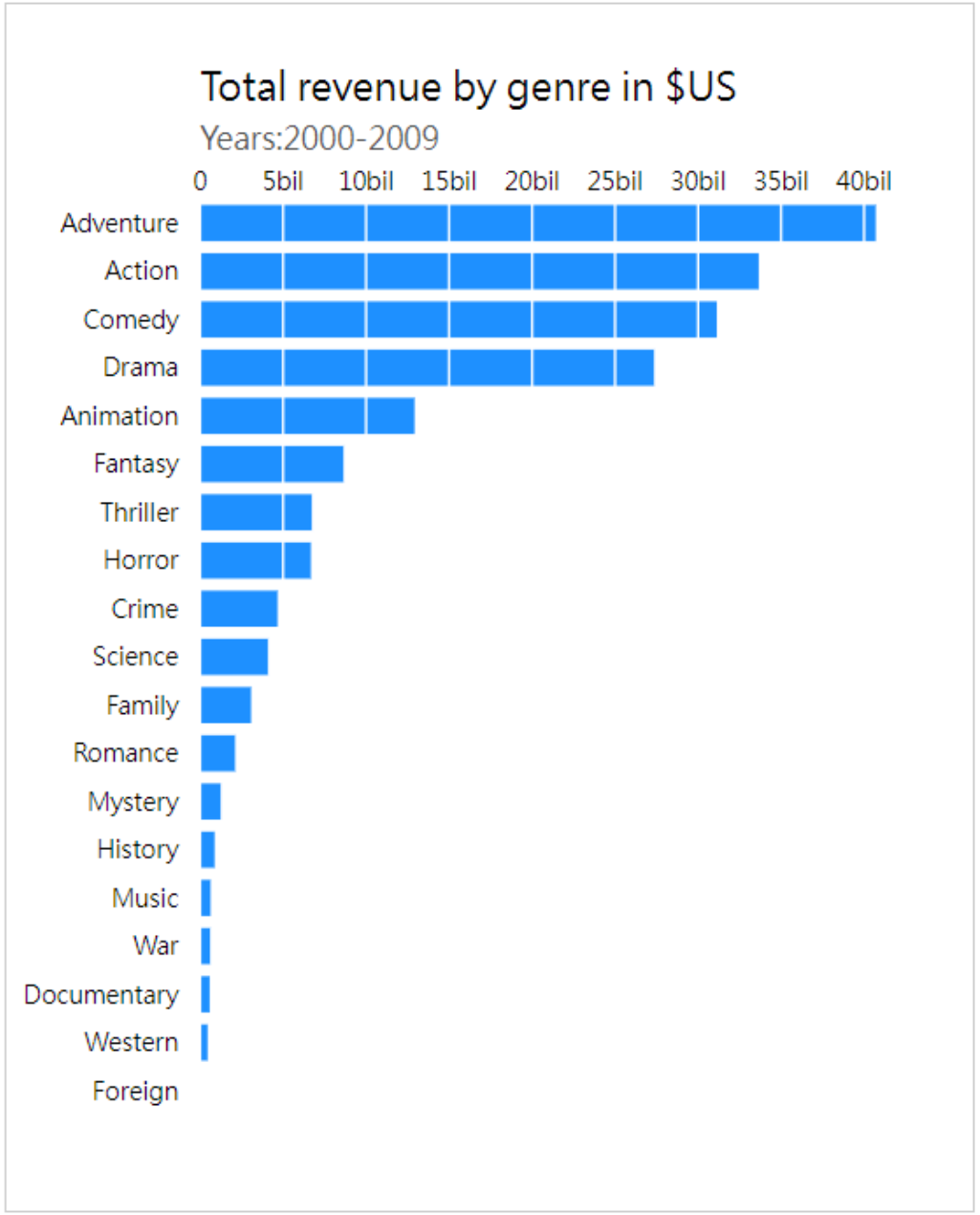
長條圖

電影分類 VS. 收益



資料處理流程

- d3.csv
- d3.json

- 缺失值處理
- 字串->數字
- 字串->日期格式
- 取得物件部分內容

- 條件過濾

main.js

資料取得 → 資料前處理 → 資料篩選

```
//Load Data
d3.csv('data/movies.csv').then(
    res=>{
        console.log(res);
    }
)
```

```
                                                                                          main.js:62
(23293) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
▶ {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, …]
>
```

main.js

資料取得 → 資料前處理 → 資料篩選

```
//Data utilities
//遇到NA就設定為undefined, 要不然就維持原本的字串
const parseNA = string => (string === 'NA' ? undefined : string);
//日期處理
const parseDate = string => d3.timeParse('%Y-%m-%d')(string);
```

main.js

```
// + 轉換成數字
function type(d){
    const date = parseDate(d.release_date);
    return {
        budget:+d.budget,
        genre:parseNA(d.genre),
        genres:JSON.parse(d.genres).map(d=>d.name),
        homepage:parseNA(d.homepage),
        id:+d.id,
        imdb_id:parseNA(d.imdb_id),
        original_language:parseNA(d.original_language),
        overview:parseNA(d.overview),
        popularity:+d.popularity,
        poster_path:parseNA(d.poster_path),
        production_countries:JSON.parse(d.production_countries),
        release_date:date,
        release_year:date.getFullYear(),
        revenue:+d.revenue,
        runtime:+d.runtime,
        tagline:parseNA(d.tagline),
        title:parseNA(d.title),
        vote_average:+d.vote_average,
        vote_count:+d.vote_count,
    }
}
```

main.js

資料取得 → 資料前處理 → 資料篩選

```
//Load Data
d3.csv('data/movies.csv',type).then(
    res=>{
        console.log(res);
    }
)
```

```
budget: 42150098
▶ genre: ƒ String()
▶ genres: (3) ['Animation', 'Comedy', 'Family']
▶ homepage: ƒ String()
id: 862
▶ imdb_id: ƒ String()
▶ original_language: ƒ String()
▶ overview: ƒ String()
popularity: 21.946943
▶ poster_path: ƒ String()
▶ production_countries: [{…}]
▶ release_date: Mon Oct 30 1995 00:00:00 GMT+0800 (台北標準時間) {}
release_year: 1995
revenue: 524844632
runtime: 81
tagline: undefined
▶ title: ƒ String()
vote_average: 7.7
vote_count: 5415
```

main.js

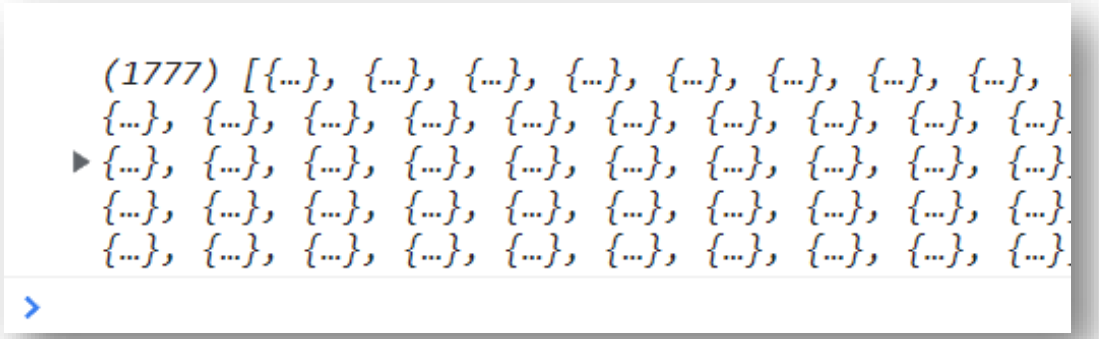
資料取得 → 資料前處理 → 資料篩選

```
//Data selection
function filterData(data){
    return data.filter(
        d => {
            return(
                d.release_year > 1999 && d.release_year < 2010 &&
                d.revenue > 0 &&
                d.budget > 0 &&
                d.genre &&
                d.title
            );
        }
    );
}
```

main.js

資料取得 → 資料前處理 → 資料篩選

```
//Main
function ready(movies){
    const moviesClean = filterData(movies);
    console.log(moviesClean);
}

//Load Data
d3.csv('data/movies.csv',type).then(
    res=>{
        ready(res);
        //console.log(res);
    }
)
```



資料聚合

- 依據電影類型(genre)來分類
- 將各分類收益加總(revenue)

data > movies.csv

Budget	Genre	Genres	Homepage	Revenue	Runtime	Status	Tagline	Title
42150098	Animation	[{"id": 16, "name	http://toystory.d	524844632	81	Released	NA	Toy Story
91444202	Adventure	[{"id": 12, "name	NA	369712072	104	Released	Roll the dice and	Jumanji
0	Romance	[{"id": 10749, "na	NA	0	101	Released	Still Yelling. Still	Grumpier Old M
22509342	Comedy	[{"id": 35, "name	NA	114589652	127	Released	Friends are the p	Waiting to Exhal
0	Comedy	[{"id": 35, "name	NA	109590055	106	Released	Just When His W	Father of the Bri
84410033	Action	[{"id": 28, "name	NA	263692465	170	Released	A Los Angeles C	Heat
81596365	Comedy	[{"id": 35, "name	NA	0	127	Released	You are cordially	Sabrina
0	Action	[{"id": 28, "name	NA	0	97	Released	The Original Bac	Tom and Huck
49239186	Action	[{"id": 28, "name	NA	90530001	106	Released	Terror goes into	Sudden Death
81543242	Adventure	[{"id": 12, "name	http://www.mgm	495155921	130	Released	No limits. No fea	GoldenEye
87166914	Comedy	[{"id": 35, "name	NA	151669722	106	Released	Why can't the m	The American Pr
0	Comedy	[{"id": 35, "name	NA	0	88	Released	NA	Dracula: Dead ar

main.js

```
function prepareBarChartData(data){
    console.log(data);
    const dataMap = d3.rollup(
        data,
        v => d3.sum(v, leaf => leaf.revenue), //將revenue加總
        d => d.genre //依電影分類groupby
    );
    const dataArray = Array.from(dataMap, d=>({genre:d[0], revenue:d[1]}));
    return dataArray;
}

//Main
function ready(movies){
    const moviesClean = filterData(movies);
    const barChartData = prepareBarChartData(moviesClean).sort(
        (a,b)=>{
            return d3.descending(a.revenue, b.revenue);
        }
    );
    console.log(barChartData);
}
```

```
▶ 0: {genre: 'Adventure', revenue: 40705505709}
▶ 1: {genre: 'Action', revenue: 33656636378}
▶ 2: {genre: 'Comedy', revenue: 31129104951}
▶ 3: {genre: 'Drama', revenue: 27345758650}
▶ 4: {genre: 'Animation', revenue: 12919617038}
▶ 5: {genre: 'Fantasy', revenue: 8636919781}
▶ 6: {genre: 'Thriller', revenue: 6728307118}
▶ 7: {genre: 'Horror', revenue: 6674908991}
▶ 8: {genre: 'Crime', revenue: 4666988126}
▶ 9: {genre: 'Science', revenue: 4087271070}
▶ 10: {genre: 'Family', revenue: 3074050308}
▶ 11: {genre: 'Romance', revenue: 2107651011}
▶ 12: {genre: 'Mystery', revenue: 1239817388}
▶ 13: {genre: 'History', revenue: 875778996}
▶ 14: {genre: 'Music', revenue: 628857031}
▶ 15: {genre: 'War', revenue: 602598594}
▶ 16: {genre: 'Documentary', revenue: 576217450}
▶ 17: {genre: 'Western', revenue: 455124613}
▶ 18: {genre: 'Foreign', revenue: 2720481}
  length: 19
```

準備將資料繪製至畫面上

- 在網頁上準備一個區塊
- 添加svg元件於該區塊
  - 設定svg元件的高度寬度
- 添加g 群組元件
  - 設定g群組元件的起始位置

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <div class="bar-chart-container"></div>
        <script src="//unpkg.com/d3"></script>
        <script src="main.js"></script>
    </body>
</html>
```

main.js

```
function ready(movies){
    const movieClean = filterData(movies);
    const barChartData = prepareBarChartData(movieClean).sort(
        (a,b)=>{
            return d3.descending(a.revenue, b.revenue);
        }
    );
    console.log(barChartData);
    setupCanvas(barChartData);

}
```

水平與垂直

- 水平
  - 將資料的最小值至最大值，對應至畫面的寬度
  - 將資料的0至最大值，對應至畫面的寬度
  - 簡化寫法
- 垂直
  - 將資料的種類，對應至畫面的高度

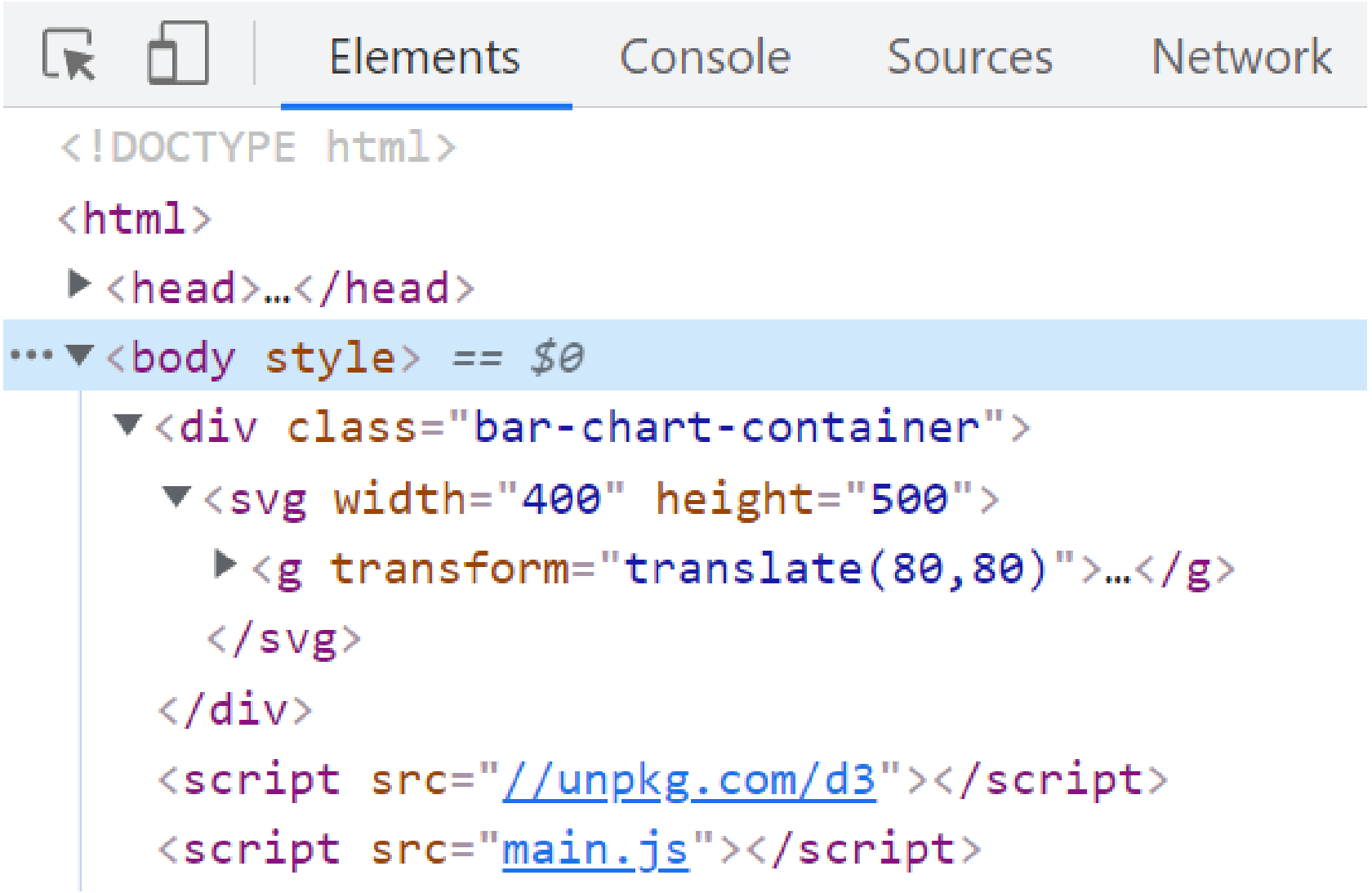
main.js - setupCanvas function

```
function setupCanvas(barChartData){
    const svg_width = 400;
    const svg_height = 500;
    const chart_margin = {top:80,right:40,bottom:40,left:80};
    const chart_width = svg_width - (chart_margin.left + chart_margin.right);
    const chart_height = svg_height - (chart_margin.top + chart_margin.bottom);

    const this_svg = d3.select('.bar-chart-container').append('svg')
      .attr('width', svg_width).attr('height',svg_height)
      .append('g')
      .attr('transform',`translate(${chart_margin.left},${chart_margin.top})`);

    //scale
    //V1.d3.extent find the max & min in revenue
    const xExtent = d3.extent(barChartData, d=>d.revenue);
    const xScale_v1 = d3.scaleLinear().domain(xExtent).range([0,chart_width]);
    //V2.0 ~ max
    const xMax = d3.max(barChartData, d=>d.revenue);
    const xScale_v2 = d3.scaleLinear().domain([0, xMax]).range([0,chart_width]);
    //V3.Short writing for v2
    const xScale_v3 = d3.scaleLinear([0,xMax],[0, chart_width]);
```

main.js - setupCanvas function

```
    //垂直空間的分配 - 平均分布給各種類
    const yScale = d3.scaleBand().domain(barChartData.map(d=>d.genre))
                     .rangeRound([0, chart_height])
                     .paddingInner(0.25);

}
```

確認畫面上是否有該元件

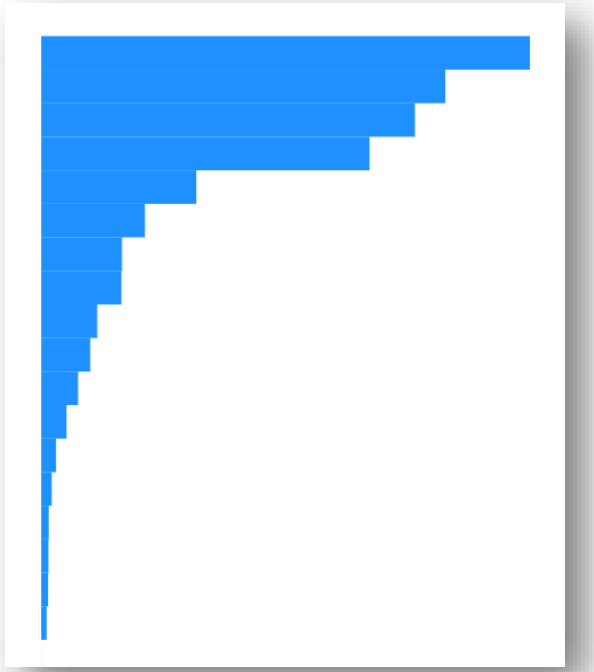


顯示 Bar圖

- 在svg中
  - 設定資料來源
  - 設定每一條Bar
    - 座標x
    - 座標y
    - 寬度
    - 高度
    - 樣式

main.js - setupCanvas

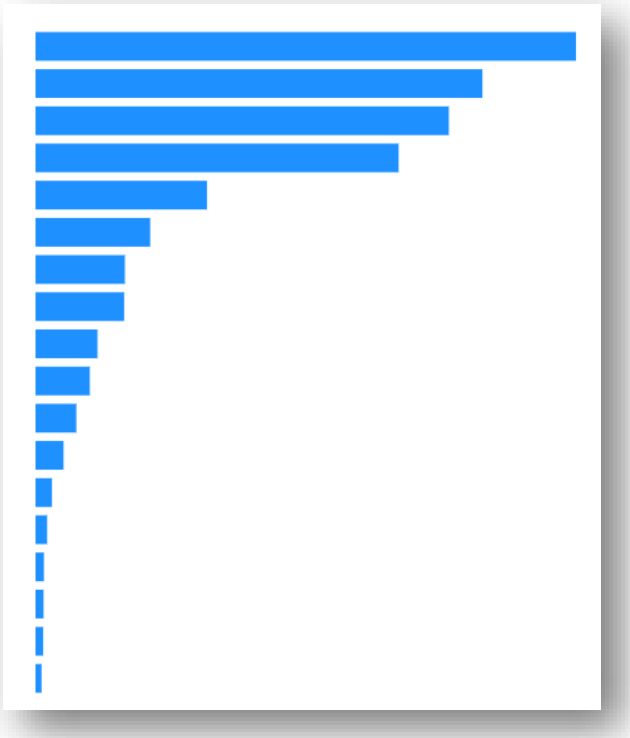
```
//Draw bars
const bars = this_svg.selectAll('.bar')
                     .data(barChartData)
                     .enter()
                     .append('rect')
                     .attr('class','bar')
                     .attr('x',0)
                     .attr('y',d=>yScale(d.genre))
                     .attr('width',d=>xScale_v3(d.revenue))
                     .attr('height',yScale.bandwidth())
                     .style('fill','dodgerblue')
```



main.js - ready function

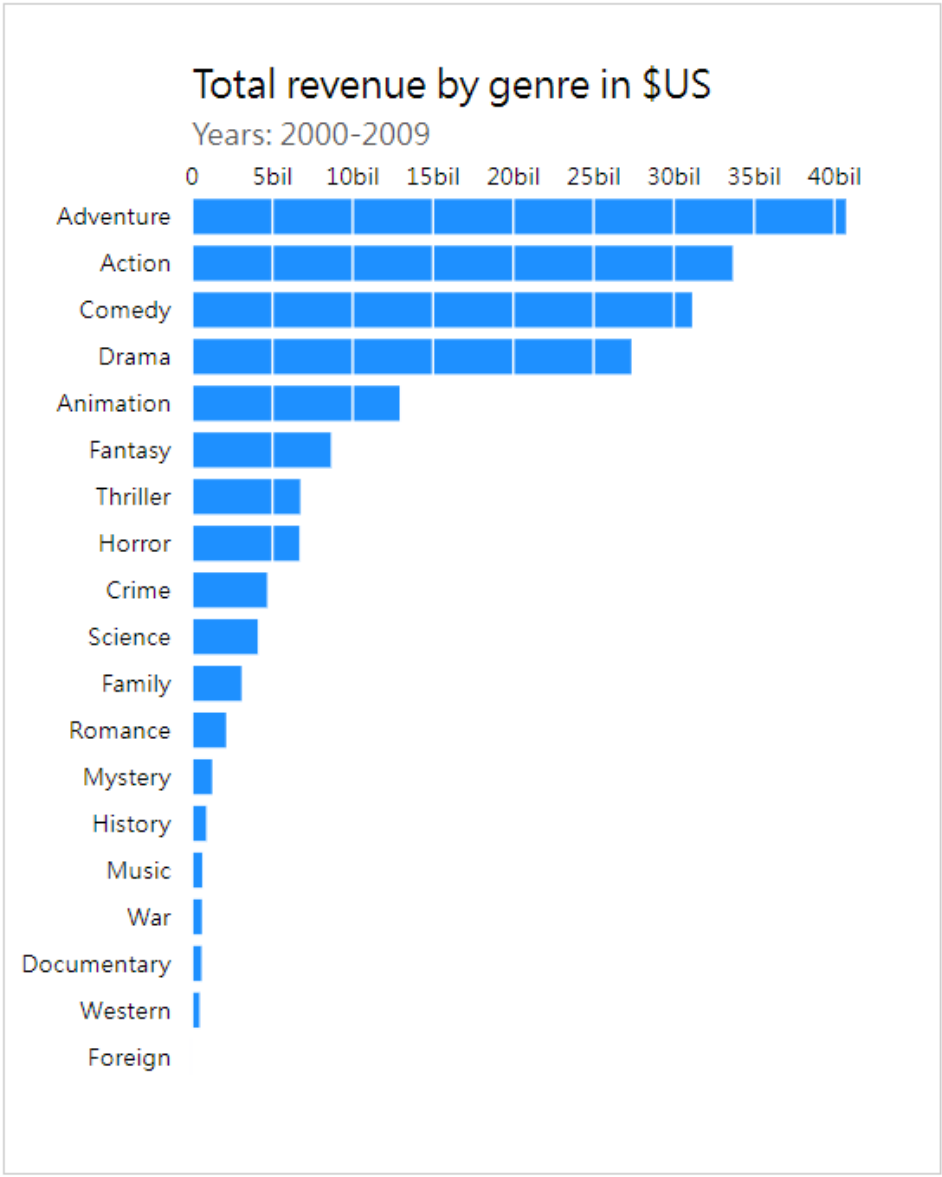
- 增加bar之間的空隙

```
//顯示電影種類(垂直) - 將垂直空間平均分布給各種類
    const yScale = d3.scaleBand().domain(barChartData.map(d=>d.genre))
                     .rangeRound([0, chart_height])
                     .paddingInner(0.25);
```



加上標題

- 主題
- X軸
- Y軸


Total revenue by genre in $US
Years: 2000-2009
0
5bil
10bil
15bil
20bil
25bil
30bil
35bil
40bil
Adventure
Action
Comedy
Drama
Animation
Fantasy
Thriller
Horror
Crime
Science
Family
Romance
Mystery
History
Music
War
Documentary
Western
Foreign

main.js

```
//Draw header
const header = this_svg.append('g').attr('class','bar-header')
                .attr('transform',`translate(0,${-chart_margin.top/2})`)
                .append('text');
header.append('tspan').text('Total revenue by genre in $US');
header.append('tspan').text('Years:2000-2009')
      .attr('x',0).attr('y',20).style('font-size','0.8em').style('fill','#555');
```

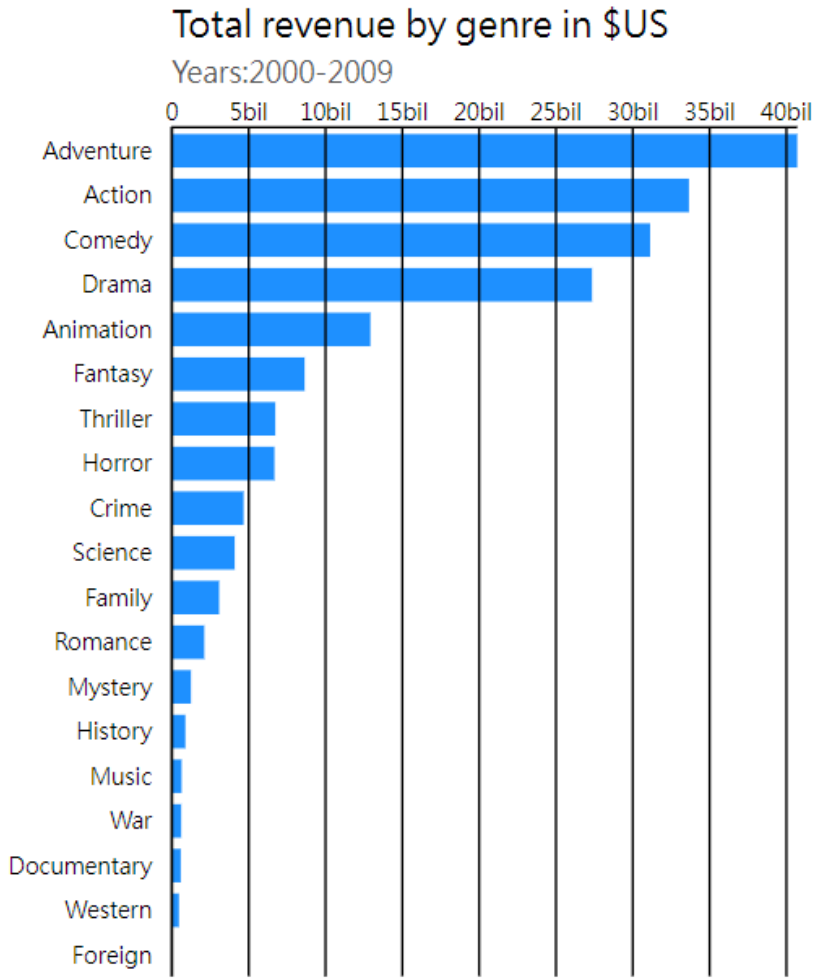
main.js

- 刻度顯示格式轉換

```
function formatTicks(d){
    return d3.format('~s')(d)
    .replace('M','mil')
    .replace('G','bil')
    .replace('T','tri')
}
```

main.js

```
//tickSizeInner : the length of the tick lines
//tickSizeOuter : the length of the square ends of the domain path
const xAxis = d3.axisTop(xScale_v3)
                .tickFormat(formatTicks)
                .tickSizeInner(-chart_height)
                .tickSizeOuter(0);
const xAxisDraw = this_svg.append('g')
                          .attr('class','x axis')
                          .call(xAxis);

const yAxis = d3.axisLeft(yScale).tickSize(0);
const yAxisDraw = this_svg.append('g')
                          .attr('class','y axis')
                          .call(yAxis);
yAxisDraw.selectAll('text').attr('dx','-0.6em');
```



style.css

```
svg{
    border: 1px solid #ccc;
}

.axis .domain{
    stroke:none;
}

.x line{
    stroke: white;
}
```