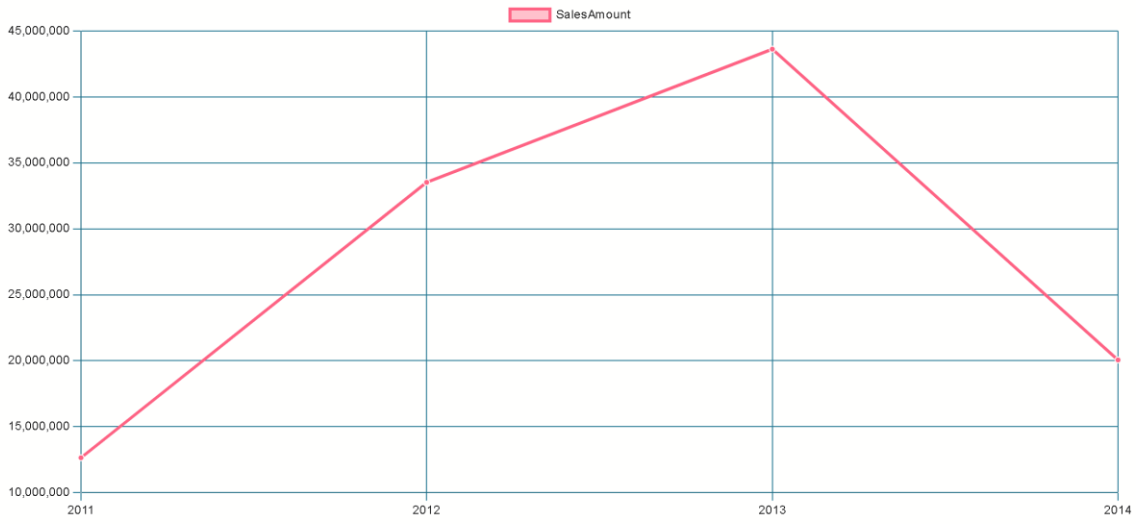
This project analyzes sales data from the AdventureWorks2019 sample database with a focus on the Bikes product category. The analysis queries *Sales.SalesOrderHeader* and *Sales.SalesOrderDetail* to measure revenue trends over time and joins to *Production.Product*, *ProductSubcategory*, and *ProductCategory* to segment revenue by category.

Key questions:

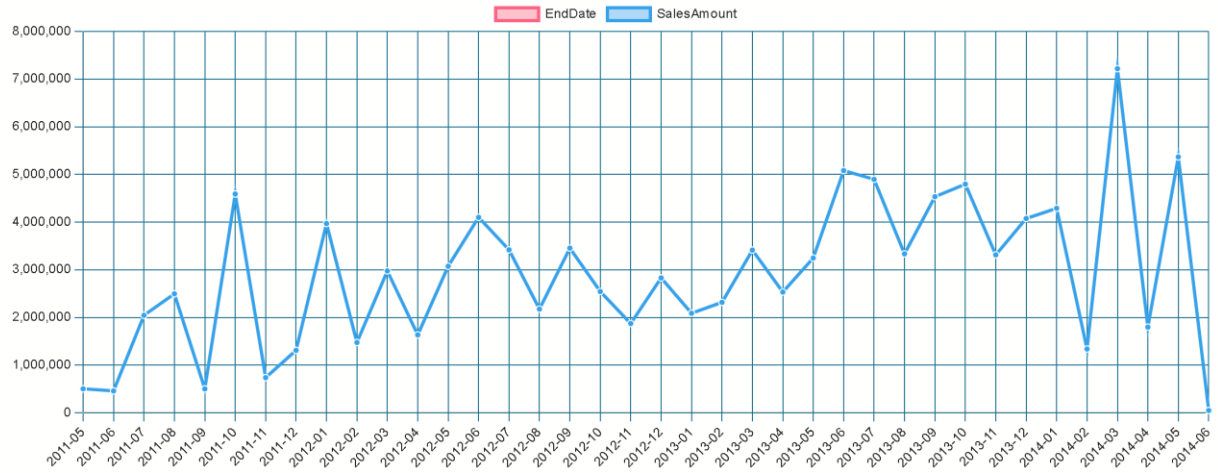
- How does total sales revenue change by year and by month?
- How does Bikes revenue change by year and by month?
- How does Bikes performance compare with other product categories?
- What share of revenue comes from Bikes versus all other categories (yearly and monthly)?

```
USE AdventureWorks2019
GO

/* 1. Total Sales by time */

    -- Total sales by years
SELECT FORMAT(OrderDate, 'yyyy') as Year, ROUND(SUM(LineTotal), 0) as SalesAmount
FROM Sales.SalesOrderDetail d
JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
GROUP BY FORMAT(OrderDate, 'yyyy')
ORDER BY Year
```

	Year	SalesAmount
1	2011	12641672.000000
2	2012	33524301.000000
3	2013	43622479.000000
4	2014	20057929.000000

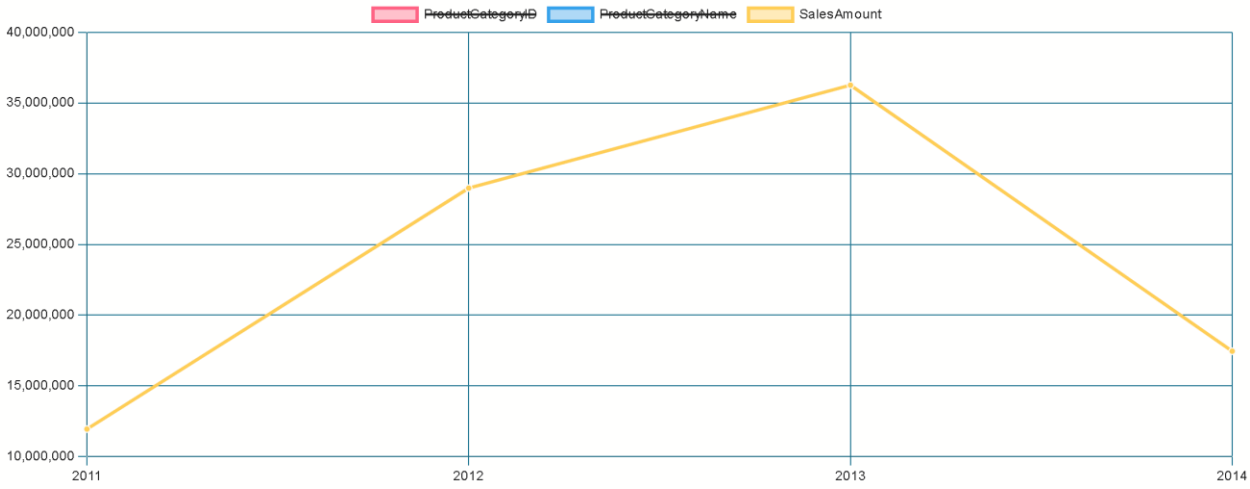



```
    -- Total sales by months
SELECT FORMAT(OrderDate, 'yyyy-MM') as YearMonth, EOMONTH(OrderDate) AS EndDate,
ROUND(SUM(LineTotal), 0) as SalesAmount
FROM Sales.SalesOrderDetail d
JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
GROUP BY FORMAT(OrderDate, 'yyyy-MM'), EOMONTH(OrderDate)
ORDER BY YearMonth, EndDate
```



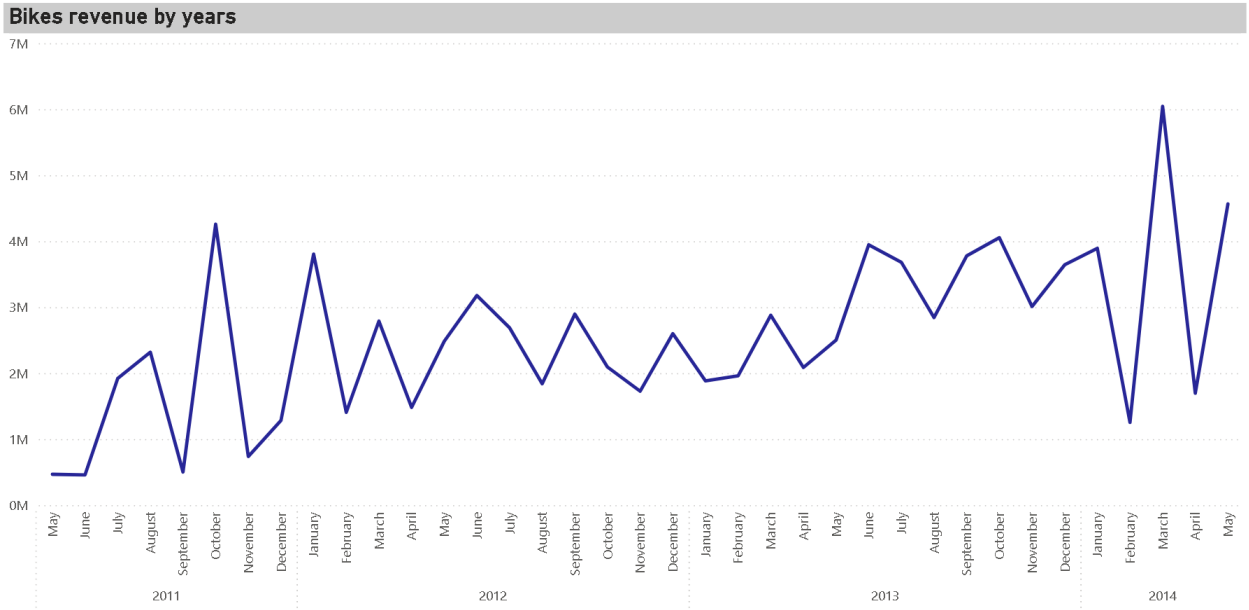

```sql
/* 2. 'Bikes' sales by time series */

    -- 'Bikes' revenue over years
SELECT FORMAT(OrderDate, 'yyyy') as Year, c.ProductCategoryID, c.Name as ProductCategoryName,
ROUND(SUM(LineTotal), 0) as SalesAmount
FROM Sales.SalesOrderDetail d
JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
JOIN Production.Product p on d.ProductID = p.ProductID
JOIN Production.ProductSubcategory sc on p.ProductSubcategoryID = sc.ProductSubcategoryID
JOIN Production.ProductCategory c on sc.ProductCategoryID = c.ProductCategoryID
WHERE c.Name = 'Bikes'
GROUP BY FORMAT(OrderDate, 'yyyy'), c.ProductCategoryID, c.Name
ORDER BY Year, c.ProductCategoryID
```

	Year	ProductCategoryID	ProductCategoryName	SalesAmount
1	2011	1	Bikes	11945647.000000
2	2012	1	Bikes	28985516.000000
3	2013	1	Bikes	36266829.000000
4	2014	1	Bikes	17453181.000000

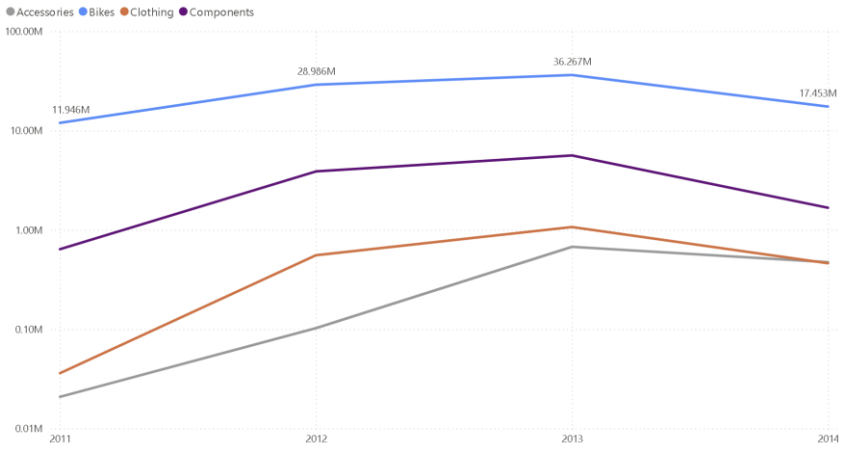



```sql
    -- 'Bikes' revenue over months
SELECT FORMAT(OrderDate, 'yyyy-MM') as YearMonth, EOMONTH(OrderDate) AS EndDate, c.ProductCategoryID, c.Name
as ProductCategoryName,
ROUND(SUM(LineTotal), 0) as SalesAmount
FROM Sales.SalesOrderDetail d
JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
```

```
JOIN Production.Product p on d.ProductID = p.ProductID
JOIN Production.ProductSubcategory sc on p.ProductSubcategoryID = sc.ProductSubcategoryID
JOIN Production.ProductCategory c on sc.ProductCategoryID = c.ProductCategoryID
WHERE c.Name = 'Bikes'
GROUP BY FORMAT(OrderDate, 'yyyy-MM'), EOMONTH(OrderDate), c.ProductCategoryID, c.Name
ORDER BY YearMonth, c.ProductCategoryID
```



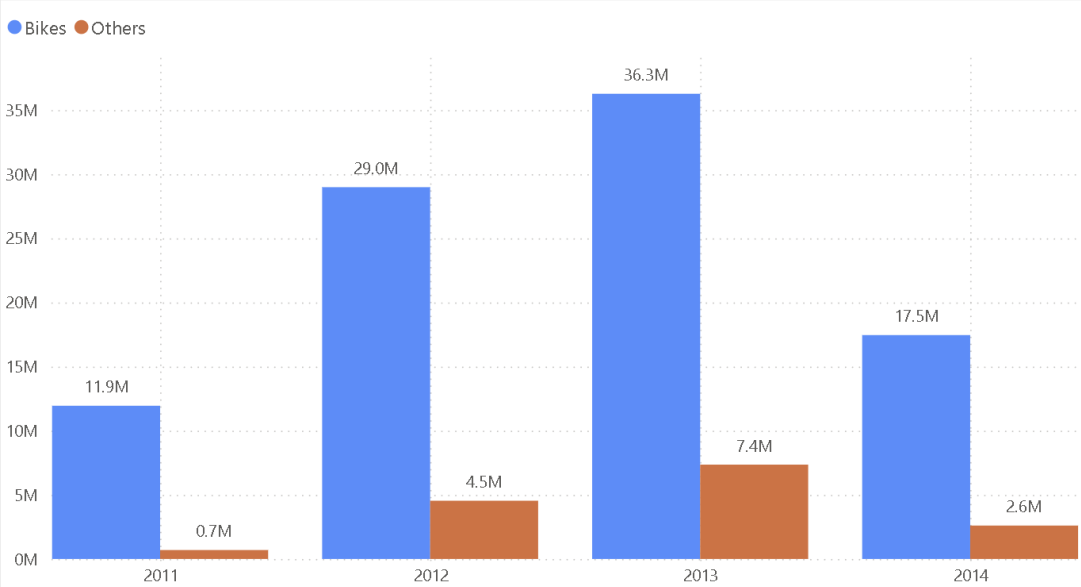

```
/* 3. 'Bikes' versus over categories */

    -- All categories over years
SELECT FORMAT(OrderDate, 'yyyy') as Year, c.ProductCategoryID, c.Name as ProductCategoryName,
CONVERT(DECIMAL(18,0), SUM(LineTotal)) as SalesAmount
FROM Sales.SalesOrderDetail d
JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
JOIN Production.Product p on d.ProductID = p.ProductID
JOIN Production.ProductSubcategory sc on p.ProductSubcategoryID = sc.ProductSubcategoryID
JOIN Production.ProductCategory c on sc.ProductCategoryID = c.ProductCategoryID
GROUP BY FORMAT(OrderDate, 'yyyy'), c.ProductCategoryID, c.Name
ORDER BY Year, c.ProductCategoryID, c.Name
```




```
    -- 'Bikes' versus others
SELECT [Year], ProductCategoryID, ProductCategoryName, CONVERT(DECIMAL(18,0), SUM(LineTotal)) as SalesAmount
FROM
(SELECT FORMAT(OrderDate, 'yyyy') as Year,
    CASE WHEN c.Name='Bikes' THEN c.ProductCategoryID ELSE 0 END AS ProductCategoryID,
```

```
    CASE WHEN c.Name='Bikes' THEN c.Name ELSE 'Others' END AS ProductCategoryName,
    LineTotal
    FROM Sales.SalesOrderDetail d
    JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
    JOIN Production.Product p on d.ProductID = p.ProductID
    JOIN Production.ProductSubcategory sc on p.ProductSubcategoryID = sc.ProductSubcategoryID
    JOIN Production.ProductCategory c on sc.ProductCategoryID = c.ProductCategoryID
) s
GROUP BY Year, ProductCategoryID, ProductCategoryName
ORDER BY Year, ProductCategoryID
```

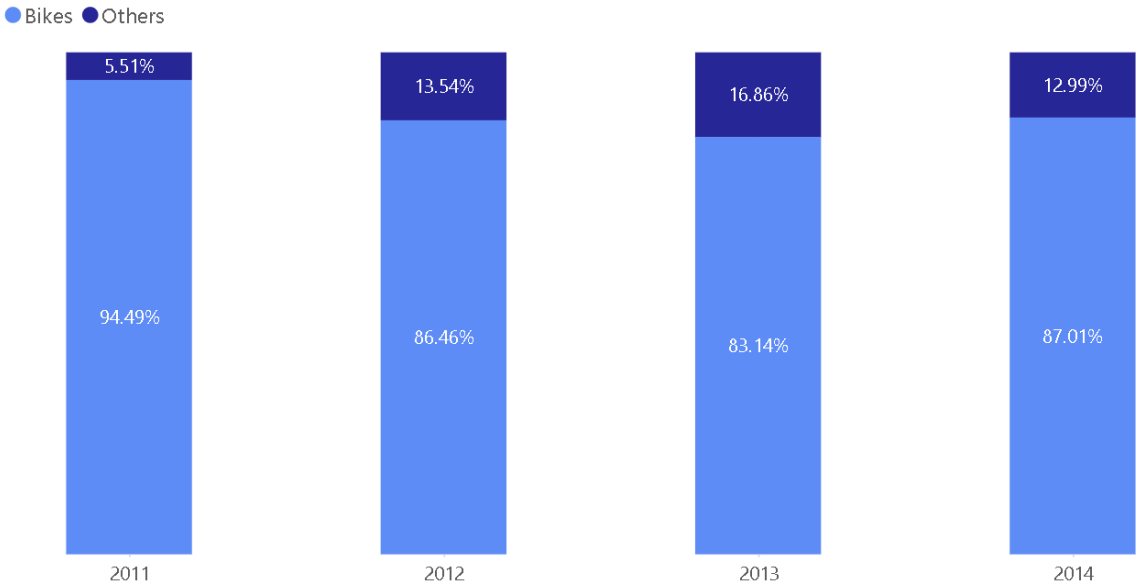
	Year	ProductCategoryID	ProductCategoryName	SalesAmount
1	2011	0	Others	696025
2	2011	1	Bikes	11945647
3	2012	0	Others	4538785
4	2012	1	Bikes	28985516
5	2013	0	Others	7355650
6	2013	1	Bikes	36266829
7	2014	0	Others	2604748
8	2014	1	Bikes	17453181




```
    -- 'Bikes' distribution by years
;WITH s AS (
SELECT [Year], ProductCategoryID, ProductCategoryName, CONVERT(DECIMAL(18,0), SUM(LineTotal)) as SalesAmount
    FROM
    (SELECT FORMAT(OrderDate, 'yyyy') as Year,
        CASE WHEN c.Name='Bikes' THEN c.ProductCategoryID ELSE 0 END AS ProductCategoryID,
        CASE WHEN c.Name='Bikes' THEN c.Name ELSE 'Others' END AS ProductCategoryName,
        LineTotal
        FROM Sales.SalesOrderDetail d
        JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
        JOIN Production.Product p on d.ProductID = p.ProductID
        JOIN Production.ProductSubcategory sc on p.ProductSubcategoryID = sc.ProductSubcategoryID
        JOIN Production.ProductCategory c on sc.ProductCategoryID = c.ProductCategoryID
    ) s
    GROUP BY Year, ProductCategoryID, ProductCategoryName
    -- ORDER BY Year, ProductCategoryID
), yearly AS (
```

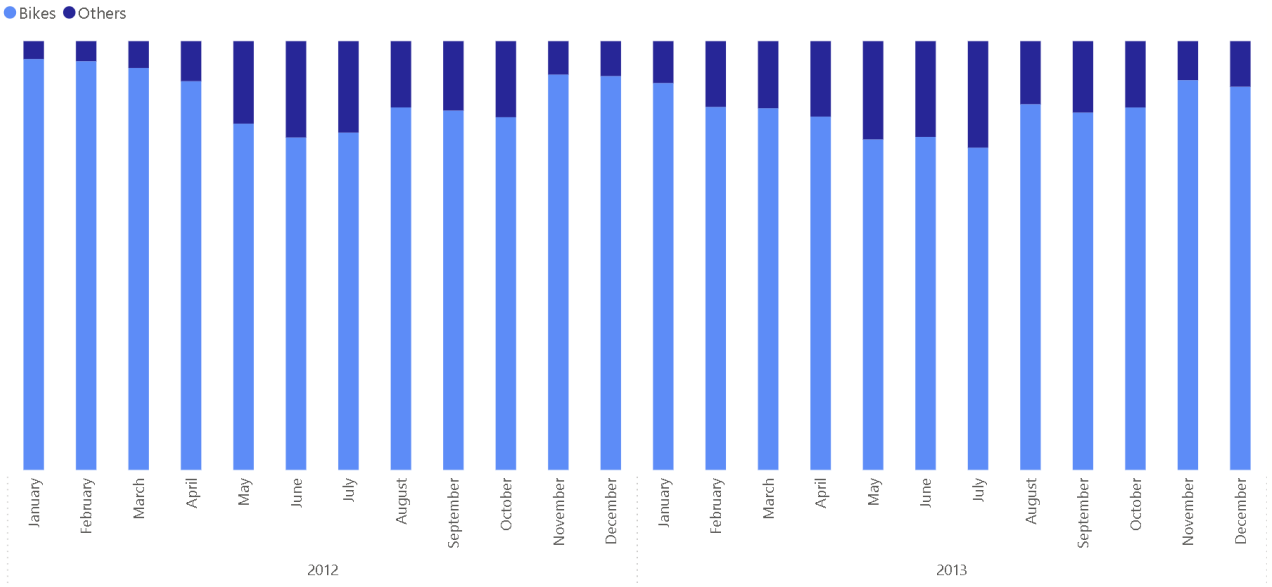
```
SELECT [Year],
    SUM(CASE WHEN ProductCategoryName='Bikes' THEN SalesAmount ELSE 0 END) AS Bikes,
    SUM(CASE WHEN ProductCategoryName='Others' THEN SalesAmount ELSE 0 END) AS Others,
    SUM(SalesAmount) AS [Total]
    FROM s
    GROUP BY [Year]
)

SELECT [Year], Bikes - Others AS Difference, ROUND(Bikes / Others, 1) AS Ratio,
ROUND(Bikes * 100 / Total, 2) AS BikesPercentOfTotal, ROUND(Others * 100 / Total, 2) AS OthersPercentOfTotal
FROM yearly
ORDER BY [YEAR]
```

	Year	Difference	Ratio	BikesPercentOfTotal	OthersPercentOfTotal
1	2011	11249622	17.200000	94.490000	5.510000
2	2012	24446731	6.400000	86.460000	13.540000
3	2013	28911179	4.900000	83.140000	16.860000
4	2014	14848433	6.700000	87.010000	12.990000




```
    -- 'Bikes' distribution by months
;WITH s AS (
SELECT YearMonth, EndDate, ProductCategoryID, ProductCategoryName, CONVERT(DECIMAL(18,0), SUM(LineTotal)) as
SalesAmount
    FROM
    (SELECT FORMAT(OrderDate, 'yyyy-MM') as YearMonth, EOMONTH(OrderDate) AS EndDate,
        CASE WHEN c.Name='Bikes' THEN c.ProductCategoryID ELSE 0 END AS ProductCategoryID,
        CASE WHEN c.Name='Bikes' THEN c.Name ELSE 'Others' END AS ProductCategoryName,
        LineTotal
        FROM Sales.SalesOrderDetail d
        JOIN Sales.SalesOrderHeader h on h.SalesOrderID = d.SalesOrderID
        JOIN Production.Product p on d.ProductID = p.ProductID
        JOIN Production.ProductSubcategory sc on p.ProductSubcategoryID = sc.ProductSubcategoryID
        JOIN Production.ProductCategory c on sc.ProductCategoryID = c.ProductCategoryID
        WHERE YEAR(OrderDate) IN (2012, 2013)
    ) s
    GROUP BY YearMonth, EndDate, ProductCategoryID, ProductCategoryName
    --ORDER BY YearMonth, EndDate, ProductCategoryID
```

```
), monthly AS (
SELECT YearMonth, EndDate,
    SUM(CASE WHEN ProductCategoryName='Bikes' THEN SalesAmount ELSE 0 END) AS Bikes,
    SUM(CASE WHEN ProductCategoryName='Others' THEN SalesAmount ELSE 0 END) AS Others,
    SUM(SalesAmount) AS [Total]
    FROM s
    GROUP BY YearMonth, EndDate
)

SELECT YearMonth, EndDate, Bikes - Others AS Difference, ROUND(Bikes / Others, 1) AS Ratio,
ROUND(Bikes * 100 / Total, 2) AS BikesPercentOfTotal, ROUND(Others * 100 / Total, 2) AS OthersPercentOfTotal
FROM monthly
ORDER BY YearMonth, EndDate
```



Overall revenue increased strongly from 2011 to 2013, then declined in 2014 (the dataset appears to be partial-year for 2014 based on the monthly timeline).

Bikes is the main revenue driver every year, contributing the majority of total sales; the yearly share ranges roughly from the low-80%s to mid-90%s.

The Bikes vs. Others gap is consistently large: Bikes revenue is multiple times higher than the combined revenue of all other categories each year.

Among non-bike categories, Components contributes the most revenue, while Clothing and Accessories remain comparatively smaller; all follow a similar pattern of growth through 2013 and softening afterward.

Monthly results show noticeable seasonality/volatility, with higher and more sustained Bikes revenue levels in 2013 and several sharp monthly spikes across the timeline.

P/s: Some of the visualizations are generated by PowerBI