

Riverbend Transit — Ridership Analysis with Python in Excel

Native formulas and Python in Excel side by side over a year of transit operations data.

Instructions

Download the starter workbook and complete each task in the exact sheet and cell indicated. Every answer must be a formula or a Python in Excel expression — a typed-in number will not receive credit even if it is correct. Do not rename, delete, or restructure any existing sheets or columns, as the autograder depends on the original layout. Save your work frequently and submit the completed .xlsx file to the course portal before the deadline.

Scenario

Tri-Valley Regional Transit Authority (TVRTA) has hired you as a junior data analyst to evaluate one quarter of daily bus operations. Management needs a clear picture of ridership patterns, fuel efficiency, and on-time performance across all four service zones before presenting findings to the regional transportation board. Your Excel workbook combines raw operational logs with route reference data so you can build the summaries, pivot analyses, and visualisations the board requires.

Workbook

Tri-Valley Regional Transit Authority (TVRTA) operates a network of bus routes across four service zones covering suburban and urban corridors. The authority monitors daily operational data to evaluate route efficiency, driver performance, and fuel consumption. Management uses this data to make scheduling decisions, allocate resources, and report on-time performance to the regional transportation board. The dataset covers a full operational quarter and supports analysis of ridership trends, fuel efficiency, and service reliability by route, vehicle, and zone.

Tasks

1. In cell B2 of the Analysis sheet, calculate the total number of passenger boardings (TotalRiders) across all 80 log entries in the Operations sheet. (5 pts)

Answer in: Analysis!B2

Solution:

=SUM(Operations!K2:K81)

SUM over all 80 TotalRiders values in column K of the Operations sheet.

2. In cell B3 of the Analysis sheet, calculate the average fuel consumption in gallons (FuelUsedGallons) across all 80 log entries in the Operations sheet. Round to 2 decimal places using the ROUND function. (5 pts)

Answer in: Analysis!B3

Solution:

`=ROUND(AVERAGE(Operations!N2:N81), 2)`

AVERAGE over all 80 FuelUsedGallons values in column N of the Operations sheet, wrapped in ROUND to 2 decimal places.

3. In cells B6 through B9 of the Analysis sheet, use SUMIF to calculate the total TotalRiders for each of the four zones. Cell B6 should hold the total for the 'North' zone, B7 for 'South', B8 for 'East', and B9 for 'West'. Use the zone labels already entered in cells A6:A9 as the criteria. Reference the Zone column (C2:C81) and TotalRiders column (K2:K81) on the Operations sheet. (10 pts)

Answer in: Analysis!B6, B7, B8, B9

Hint: Use absolute references on the Operations ranges so the formula can be filled down for all four zones.

Solution:

`=SUMIF(Operations!$C$2:$C$81, A6, Operations!$K$2:$K$81)`

SUMIF checks the Zone column on Operations for each zone label in A6:A9 and sums the corresponding TotalRiders. Absolute references on the Operations ranges allow the formula to be filled down from B6 to B9.

4. In cell B12 of the Analysis sheet, calculate the on-time performance rate for routes where an incident was reported. The on-time performance rate is defined as the sum of OnTimeTrips divided by the sum of CompletedTrips, expressed as a percentage, but only for rows where IncidentReported equals 'Yes' on the Operations sheet. Use SUMIF functions referencing Operations!P2:P81 as the criteria range, and format the result as a percentage. (10 pts)

Answer in: Analysis!B12

Hint: Think of on-time rate as a ratio: (sum of on-time trips) ÷ (sum of completed trips), filtered to incident rows only. You will need two SUMIF expressions.

Solution:

`=SUMIF(Operations!$P$2:$P$81, "Yes", Operations!$J$2:$J$81)/SUMIF(Operations!$P$2:$P$81, "Yes", Operations!$I$2:$I$81)`

Two SUMIF calls filter for rows where IncidentReported is 'Yes'. The first sums OnTimeTrips (column J); the second sums CompletedTrips (column I). Dividing gives the conditional on-time rate. The cell should be formatted as a percentage.

5. Create a PivotTable on the Analysis sheet starting at cell D2. Use the Operations sheet data (A1:P81) as the source. Configure the PivotTable with: RouteCode as Row labels, VehicleType as Column labels, and the Average of FuelUsedGallons as the Values field.

Format the values in the PivotTable to show 2 decimal places. (15 pts)

Answer in: Analysis!D2

Hint: Use Insert > PivotTable and point the data source at the full Operations table range including the header row.

Solution:

Insert a PivotTable from Operations!A1:P81. Drag RouteCode to Rows, VehicleType to Columns, and FuelUsedGallons to Values summarised as Average. Format values to 2 decimal places.

6. On the Dashboard sheet, create a Bar Chart (clustered bar) that visualises total TotalRiders by Zone. Use the zone totals you calculated in Analysis!A6:B9 as the chart data source. Title the chart 'Total Ridership by Zone'. Place the chart within the range A2:H20 on the Dashboard sheet. (15 pts)

Answer in: Dashboard

Hint: Select your zone summary data on the Analysis sheet first, then use Insert > Chart and choose Clustered Bar. Move and resize the chart onto the Dashboard sheet.

Solution:

Select Analysis!A6:B9, insert a clustered Bar Chart, set the chart title to 'Total Ridership by Zone', and position the chart in the range A2:H20 on the Dashboard sheet.

7. In cell A22 of the Dashboard sheet, write a Python in Excel formula that reads all Operations data and returns a cleaned summary DataFrame showing, for each RouteCode, the total CompletedTrips, total TotalRiders, and average FuelUsedGallons (rounded to 2 decimal places). The result should be a DataFrame with columns named 'RouteCode', 'CompletedTrips', 'TotalRiders', and 'AvgFuelGallons', sorted ascending by RouteCode. (15 pts)

Answer in: Dashboard!A22

Use Python in Excel (=PY) for this task.

Hint: Use xl() to load the Operations range as a DataFrame with headers=True, then use groupby and agg to compute the three metrics per route.

Solution:

```
df = xl("Operations!A1:P81", headers=True)
result = df.groupby("RouteCode", as_index=False).agg(
    CompletedTrips=("CompletedTrips", "sum"),
    TotalRiders=("TotalRiders", "sum"),
    AvgFuelGallons=("FuelUsedGallons", "mean")
)
result["AvgFuelGallons"] = result["AvgFuelGallons"].round(2)
result.sort_values("RouteCode").reset_index(drop=True)
```

xl() reads the Operations sheet into a DataFrame. groupby on RouteCode aggregates CompletedTrips and TotalRiders as sums and FuelUsedGallons as mean, then rounds and sorts. The final DataFrame is returned as a Python object to Excel.

8. In cell A42 of the Dashboard sheet, write a Python in Excel formula that computes a correlation matrix between the following numeric columns from the Operations sheet:

ScheduledTrips, CompletedTrips, TotalRiders, FuelUsedGallons, and AvgSpeedMPH. Return the correlation matrix as a DataFrame rounded to 3 decimal places. (10 pts)

Answer in: Dashboard!A42

Use Python in Excel (=PY) for this task.

Hint: Load the Operations data with xl(), select only the five relevant columns by name, and call the pandas method that computes pairwise correlations.

Solution:

```
df = xl("Operations!A1:P81", headers=True)
cols = ["ScheduledTrips", "CompletedTrips", "TotalRiders", "FuelUsedGallons", "AvgSpeedMPH"]
df[cols].corr().round(3)
```

xl() reads the Operations data into a DataFrame. Subsetting to the five numeric columns and calling .corr() produces a 5x5 Pearson correlation matrix. round(3) formats output to 3 decimal places.

9. In cell A55 of the Dashboard sheet, write a Python in Excel formula that produces a grouped bar chart image showing average TotalRiders by Zone and VehicleType. Use the Operations sheet data as the source. The chart should have a title of 'Avg Ridership by Zone and Vehicle Type', labelled axes (x-axis: 'Zone', y-axis: 'Average Riders'), and a legend. Return the matplotlib figure so Excel renders it as an embedded image. (15 pts)

Answer in: Dashboard!A55

Use Python in Excel (=PY) for this task.

Hint: Group the data by two categorical columns and unstack one of them to create a wide DataFrame suitable for a grouped bar chart. Return the figure object, not plt.show().

Solution:

```
df = xl("Operations!A1:P81", headers=True)
pivot = df.groupby(["Zone", "VehicleType"])["TotalRiders"].mean().unstack()
fig, ax = plt.subplots(figsize=(8, 5))
pivot.plot(kind="bar", ax=ax)
ax.set_title("Avg Ridership by Zone and Vehicle Type")
ax.set_xlabel("Zone")
ax.set_ylabel("Average Riders")
ax.legend(title="VehicleType")
plt.tight_layout()
fig
```

xl() loads Operations into a DataFrame. groupby on Zone and VehicleType computes mean TotalRiders, unstacked into a wide pivot. DataFrame.plot(kind='bar') on the pivot draws a grouped bar chart. Title, axis labels, and legend are set before returning the figure for Excel to render as an image.

Answer Key

Task 1 (5 pts) — *sum*

Location: Analysis!B2

=SUM(Operations!K2:K81)

SUM over all 80 TotalRiders values in column K of the Operations sheet.

Task 2 (5 pts) — *average*

Location: Analysis!B3

=ROUND(AVERAGE(Operations!N2:N81), 2)

AVERAGE over all 80 FuelUsedGallons values in column N of the Operations sheet, wrapped in ROUND to 2 decimal places.

Task 3 (10 pts) — *sumif*

Location: Analysis!B6, B7, B8, B9

=SUMIF(Operations!\$C\$2:\$C\$81, A6, Operations!\$K\$2:\$K\$81)

SUMIF checks the Zone column on Operations for each zone label in A6:A9 and sums the corresponding TotalRiders. Absolute references on the Operations ranges allow the formula to be filled down from B6 to B9.

Task 4 (10 pts) — *sumif, average*

Location: Analysis!B12

=SUMIF(Operations!\$P\$2:\$P\$81, "Yes", Operations!\$J\$2:\$J\$81)/SUMIF(Operations!\$P\$2:\$P\$81, "Yes", Operations!\$I\$2:\$I\$81)

Two SUMIF calls filter for rows where IncidentReported is 'Yes'. The first sums OnTimeTrips (column J); the second sums CompletedTrips (column I). Dividing gives the conditional on-time rate. The cell should be formatted as a percentage.

Task 5 (15 pts) — *pivot_table*

Location: Analysis!D2

Insert a PivotTable from Operations!A1:P81. Drag RouteCode to Rows, VehicleType to Columns, and FuelUsedGallons to Values summarised as Average. Format values to 2 decimal places.

Task 6 (15 pts) — *charts*

Location: Dashboard

Select Analysis!A6:B9, insert a clustered Bar Chart, set the chart title to 'Total Ridership by Zone', and position the chart in the range A2:H20 on the Dashboard sheet.

Task 7 (15 pts) — *py_basics, py_dataframe*

Location: Dashboard!A22

```
df = xl("Operations!A1:P81", headers=True)
result = df.groupby("RouteCode", as_index=False).agg(
    CompletedTrips=("CompletedTrips", "sum"),
    TotalRiders=("TotalRiders", "sum"),
    AvgFuelGallons=("FuelUsedGallons", "mean")
)
result["AvgFuelGallons"] = result["AvgFuelGallons"].round(2)
result.sort_values("RouteCode").reset_index(drop=True)
```

xl() reads the Operations sheet into a DataFrame. groupby on RouteCode aggregates CompletedTrips and TotalRiders as sums and FuelUsedGallons as mean, then rounds and sorts. The final DataFrame is returned as a Python object to Excel.

Task 8 (10 pts) — *py_aggregation, py_statistics*

Location: Dashboard!A42

```
df = xl("Operations!A1:P81", headers=True)
cols = ["ScheduledTrips", "CompletedTrips", "TotalRiders", "FuelUsedGallons", "AvgSpeedMPH"]
df[cols].corr().round(3)
```

xl() reads the Operations data into a DataFrame. Subsetting to the five numeric columns and calling .corr() produces a 5x5 Pearson correlation matrix. round(3) formats output to 3 decimal places.

Task 9 (15 pts) — *py_visualization*

Location: Dashboard!A55

```
df = xl("Operations!A1:P81", headers=True)
pivot = df.groupby(["Zone", "VehicleType"])["TotalRiders"].mean().unstack()
fig, ax = plt.subplots(figsize=(8, 5))
pivot.plot(kind="bar", ax=ax)
ax.set_title("Avg Ridership by Zone and Vehicle Type")
ax.set_xlabel("Zone")
ax.set_ylabel("Average Riders")
ax.legend(title="VehicleType")
plt.tight_layout()
fig
```

xl() loads Operations into a DataFrame. groupby on Zone and VehicleType computes mean TotalRiders, unstacked into a wide pivot. DataFrame.plot(kind='bar') on the pivot draws a grouped bar chart. Title, axis labels, and legend are set before returning the figure for Excel to render as an image.