

Campus Energy Audit — Analysis with Python in Excel

Mixed native-formula and Python in Excel analysis of campus building energy consumption.

Instructions

Download the starter workbook (EnergyReadings_Starter.xlsx) and complete each of the nine tasks in the exact sheet and cell specified. All answers must be live formulas or Python in Excel cells — typing a static number will receive zero credit even if the value is correct. Tasks 1–6 use native Excel functions; Tasks 7–9 use Python in Excel (available via the Formulas ribbon on supported Microsoft 365 builds). Save your completed file as LastName_FirstName_Energy.xlsx and upload it to the course portal before the deadline.

Scenario

The university sustainability office has commissioned an analysis of 120 monthly energy readings spanning six campus buildings to identify inefficiencies and prioritise retrofit spending. As a student analyst, you have been given a raw meter-reading workbook and must build summary statistics, a building-level dashboard, and weather-correlation evidence to guide the capital planning committee. Your completed workbook will be presented to senior administrators at the end of the semester.

Workbook

Monthly building-level energy and cost readings across a university campus — sized and shaped for statistical and Python in Excel work.

Tasks

1. In cell B2 of the Analysis sheet, write a formula that sums all values in the ElectricityKWh column (Readings!F2:F121) to find the total electricity consumed across all buildings and months. (5 pts)

Answer in: Analysis!B2

Solution:

`=SUM(Readings!F2:F121)`

SUM over the entire ElectricityKWh column in the Readings sheet (rows 2 through 121, covering all 120 data rows).

2. In cell B3 of the Analysis sheet, write a formula that calculates the average monthly TotalCost (Readings!J2:J121) across all buildings and months. (5 pts)

Answer in: Analysis!B3

Solution:

=AVERAGE(Readings!J2:J121)

AVERAGE over the TotalCost column in the Readings sheet to find the mean monthly cost per reading.

3. In cells C8 through C13 of the Analysis sheet, first type the six building names in B8:B13 (Wells Hall, Engineering, Library, Union, Chemistry, Recreation — one per row). Then in C8, write a SUMIF formula that totals ElectricityKWh (Readings!F2:F121) where the Building column (Readings!C2:C121) matches the name in B8. Copy the formula down through C13 so each row shows that building's total electricity consumption. (10 pts)

Answer in: Analysis!C8, C9, C10, C11, C12, C13

Hint: Use absolute references on the Readings ranges so they do not shift when you fill the formula down. The building name in column B should remain a relative reference.

Solution:

=SUMIF(Readings!\$C\$2:\$C\$121,B8,Readings!\$F\$2:\$F\$121)

SUMIF checks the Building column for a match to the label in column B, then sums the corresponding ElectricityKWh values. Absolute references on the Readings ranges allow the formula to be filled down without shifting the lookup range.

4. In cell B16 of the Analysis sheet, write a SUMIF formula that sums TotalCost (Readings!J2:J121) only for rows where BuildingType (Readings!D2:D121) equals "Academic". This will reveal how much of the total utility budget is spent on academic buildings. (10 pts)

Answer in: Analysis!B16

Hint: The criteria argument should be the text string "Academic" (with quotes inside the formula). Remember SUMIF takes the range to check, the criteria, and then the range to sum — in that order.

Solution:

=SUMIF(Readings!D2:D121,"Academic",Readings!J2:J121)

SUMIF filters the TotalCost column to only add values where the BuildingType column contains 'Academic'.

5. In the Analysis sheet, beginning at cell E2, create a PivotTable sourced from Readings!A1:J121. Configure it so that Building names appear as Row Labels, the column field is empty, and the Values area shows the Sum of ElectricityKWh and the Sum of GasTherms — one column for each measure. This table will become the data foundation for the sustainability dashboard. (15 pts)

Answer in: Analysis!E2

Hint: Insert the PivotTable from the Insert ribbon. Drag 'Building' to the Rows area, then drag 'ElectricityKWh' and 'GasTherms' to the Values area. Make sure both are set to Sum, not Count.

Solution:

A PivotTable with Building as the row field and Sum of ElectricityKWh plus Sum of GasTherms as value fields summarises total energy consumption by building in a compact two-column layout.

6. In the Dashboard sheet, insert a Bar Chart (clustered) that visualises total ElectricityKWh and total GasTherms per building, using the PivotTable data you created in Task 5 (Analysis!E2 and surrounding cells) as its source. The chart should have a descriptive title such as 'Energy Consumption by Building'. Place and resize the chart so it fits neatly within the Dashboard sheet. (15 pts)

Answer in: Dashboard

Hint: Select the building names and both energy columns in your PivotTable summary, then use Insert > Charts > Bar to create a clustered bar chart. Right-click the chart title to rename it.

Solution:

A clustered bar chart with one series for ElectricityKWh and one for GasTherms, grouped by building name, lets stakeholders instantly compare electricity versus gas use across all six buildings.

7. In cell A2 of the Analysis sheet, write a Python in Excel formula that reads the entire Readings data (Readings!A1:J121) as a DataFrame with headers, then returns a new DataFrame containing only the columns Building, ElectricityKWh, GasTherms, and TotalCost. The resulting object will spill into surrounding cells and serve as a clean reference table for further analysis. (10 pts)

Answer in: Analysis!A2

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

Hint: Use xl() with headers=True to load the data. Select multiple columns from a DataFrame by passing a list of column name strings inside double brackets.

Solution:

```
df = xl("Readings!A1:J121", headers=True)
df[["Building", "ElectricityKWh", "GasTherms", "TotalCost"]]
```

xl() reads the Readings range into a pandas DataFrame. Column selection with a list of column names produces a new four-column DataFrame that Excel spills as a Python object.

8. In cell A25 of the Analysis sheet, write a Python in Excel formula that loads Readings!A1:J121 as a DataFrame, computes the Pearson correlation coefficient between HeatingDegreeDays and GasTherms and the Pearson correlation between CoolingDegreeDays and ElectricityKWh, and returns a two-row DataFrame with columns 'Comparison' and 'Correlation' labelling each result. This will tell sustainability managers how strongly weather drives energy use. (15 pts)

Answer in: Analysis!A25

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

Hint: Use the pandas Series .corr() method to get the Pearson coefficient. Build a small DataFrame from dictionaries with two keys to return both results cleanly.

Solution:

```
df = xl("Readings!A1:J121", headers=True)
r_heat = df["HeatingDegreeDays"].corr(df["GasTherms"])
r_cool = df["CoolingDegreeDays"].corr(df["ElectricityKWh"])
pd.DataFrame({"Comparison": ["HDD vs GasTherms", "CDD vs ElectricityKWh"], "Correlation": [r_heat, r_cool]})
```

pandas .corr() computes the Pearson r between two Series. Assembling the two scalar results into a two-row DataFrame gives a clean, spill-ready output that shows the weather-energy relationship at a glance.

9. In cell A2 of the Dashboard sheet, write a Python in Excel formula that loads Readings!A1:J121 as a DataFrame, computes each building's average energy intensity as total ElectricityKWh divided by SquareFeet (aggregated per building), and produces a horizontal bar chart (using matplotlib) with buildings on the y-axis and average kWh per square foot on the x-axis. Title the chart 'Electricity Intensity by Building (kWh / sq ft)' and label the x-axis 'kWh per Square Foot'. The chart image will spill into the Dashboard sheet to highlight the worst-performing buildings. (15 pts)

Answer in: Dashboard!A2

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

Hint: Use groupby on 'Building' with an apply lambda to compute total kWh / SquareFeet for each group. Return the matplotlib figure object (fig) as the last expression so Excel renders it as an image.

Solution:

```
df = xl("Readings!A1:J121", headers=True)
intensity = df.groupby("Building").apply(lambda g: g["ElectricityKWh"].sum() / g["SquareFeet"].iloc[0])
fig, ax = plt.subplots(figsize=(8, 5))
ax.barh(intensity.index, intensity.values)
ax.set_xlabel("kWh per Square Foot")
ax.set_title("Electricity Intensity by Building (kWh / sq ft)")
plt.tight_layout()
fig
```

groupby + apply computes total kWh divided by each building's square footage to yield intensity. matplotlib barh plots a horizontal bar chart. Returning the fig object causes Excel to embed the chart as an image in the cell.

Answer Key

Task 1 (5 pts) — *sum*

Location: Analysis!B2

=SUM(Readings!F2:F121)

SUM over the entire ElectricityKWh column in the Readings sheet (rows 2 through 121, covering all 120 data rows).

Task 2 (5 pts) — *average*

Location: Analysis!B3

=AVERAGE(Readings!J2:J121)

AVERAGE over the TotalCost column in the Readings sheet to find the mean monthly cost per reading.

Task 3 (10 pts) — *sumif*

Location: Analysis!C8, C9, C10, C11, C12, C13

=SUMIF(Readings!\$C\$2:\$C\$121, B8, Readings!\$F\$2:\$F\$121)

SUMIF checks the Building column for a match to the label in column B, then sums the corresponding ElectricityKWh values. Absolute references on the Readings ranges allow the formula to be filled down without shifting the lookup range.

Task 4 (10 pts) — *sumif*

Location: Analysis!B16

=SUMIF(Readings!D2:D121, "Academic", Readings!J2:J121)

SUMIF filters the TotalCost column to only add values where the BuildingType column contains 'Academic'.

Task 5 (15 pts) — *pivot_table*

Location: Analysis!E2

A PivotTable with Building as the row field and Sum of ElectricityKWh plus Sum of GasTherms as value fields summarises total energy consumption by building in a compact two-column layout.

Task 6 (15 pts) — *charts*

Location: Dashboard

A clustered bar chart with one series for ElectricityKWh and one for GasTherms, grouped by building name, lets stakeholders instantly compare electricity versus gas use across all six

buildings.

Task 7 (10 pts) — *py_basics, py_dataframe*

Location: Analysis!A2

```
df = xl("Readings!A1:J121", headers=True)
df[["Building", "ElectricityKWh", "GasTherms", "TotalCost"]]
```

xl() reads the Readings range into a pandas DataFrame. Column selection with a list of column names produces a new four-column DataFrame that Excel spills as a Python object.

Task 8 (15 pts) — *py_aggregation, py_statistics*

Location: Analysis!A25

```
df = xl("Readings!A1:J121", headers=True)
r_heat = df["HeatingDegreeDays"].corr(df["GasTherms"])
r_cool = df["CoolingDegreeDays"].corr(df["ElectricityKWh"])
pd.DataFrame({"Comparison": ["HDD vs GasTherms", "CDD vs ElectricityKWh"], "Correlation": [r_heat, r_cool]})
```

pandas .corr() computes the Pearson r between two Series. Assembling the two scalar results into a two-row DataFrame gives a clean, spill-ready output that shows the weather-energy relationship at a glance.

Task 9 (15 pts) — *py_visualization*

Location: Dashboard!A2

```
df = xl("Readings!A1:J121", headers=True)
intensity = df.groupby("Building").apply(lambda g: g["ElectricityKWh"].sum() / g["SquareFeet"].iloc[0])
fig, ax = plt.subplots(figsize=(8, 5))
ax.barh(intensity.index, intensity.values)
ax.set_xlabel("kWh per Square Foot")
ax.set_title("Electricity Intensity by Building (kWh / sq ft)")
plt.tight_layout()
fig
```

groupby + apply computes total kWh divided by each building's square footage to yield intensity. matplotlib barh plots a horizontal bar chart. Returning the fig object causes Excel to embed the chart as an image in the cell.