Project 1 - Clinical Trial Patient Recruitment and Adherence Monitoring

Problem Statement: Pharmaceutical companies and research organizations face immense costs and delays when clinical trials fail to recruit patients on schedule. Furthermore, monitoring patient adherence to trial protocols across multiple sites is complex and often manual, risking data integrity.

Use Case: Develop a central command center for trial managers to monitor recruitment funnels and

patient adherence in real time. The solution will analyze data from Electronic Data Capture (EDC)

systems to identify recruitment bottlenecks at specific sites, forecast enrollment completion dates, and

flag patients at risk of non-adherence or dropout.

Key Modules:

• Multi-Site Data Integration Pipeline (from EDC and EMR systems).

• Patient Recruitment Funnel Visualization (Screened vs. Enrolled vs. Randomized).

• Site Performance Leaderboard (Enrollment velocity, data quality scores).

• Patient Adherence Dashboard (Visit completion, medication adherence tracking).

• Dropout Risk Prediction Model (using factors like missed visits and demographics).

1) Project overview (one-sentence)

Build a secure central command center that ingests anonymized EDC/EMR exports, models recruitment funnels and site KPIs, visualizes recruitment & adherence in Power BI, and produces a dropout-risk score per patient so trial managers can act in real time.

2) High-level architecture (components)

1. **Raw data sources** — CSV/Excel exports from EDC (Medidata/Veeva), EMR extracts, plus site metadata and trial protocol config.
2. **ETL / Ingest** — Power Query (Power BI) / Python / SQL to clean, join, anonymize and transform.
3. **Data Model** — star schema in Power BI (fact tables + dimension tables).
4. **Analytics** — DAX measures for KPIs (funnel, rates, velocities, adherence).

5. **ML** — Dropout risk prediction (train offline with Python / scikit-learn; score and import results).
6. **Visualization & UX** — Power BI dashboards: Enrollment Overview, Recruitment Funnel, Site Leaderboard, Patient Adherence, Patient Detail.
7. **Security & Ops** — Row-Level Security (RLS), scheduled refreshes, validation & audit logs.

3) Data model & sample schema

Aim for a **star schema** with a few dimension tables and a few fact tables.

Dimensions

- **DimSite**: SiteID, SiteName, Country, Region, PI_Name, Site_Manager_Email, OpeningDate, CloseDate, Site_TargetEnrollment
- **DimPatient** (anonymized): PatientPK, PseudonymID (e.g., PAT0001), Sex, DOB (or AgeBucket), EnrollmentDate, RandomizationDate, SiteID
- **DimVisitType**: VisitTypeID, VisitName, Window, ExpectedWindowDays

Fact tables

- **FactScreeningEnrollment** — one row per screening attempt
    - PatientPK, SiteID, ScreeningDate, ScreenResult (Passed/Failed), ReasonForFailure, SourceCRF
- **FactEnrollment** — enrollment events (one per patient)
    - PatientPK, SiteID, EnrollmentDate, RandomizedFlag, RandomizationDate
- **FactVisits** — visits/appointments
    - VisitID, PatientPK, SiteID, VisitDate, VisitStatus (Completed/Missed/Rescheduled), eDiarySubmitted (Y/N), MedicationTakenPercent (0-100), AE_Reported
- **FactDataQuality** — data quality events or aggregated scores per site/day
    - SiteID, Date, QueryCount, QueriesOpen, DataCompletenessPct, TimelinessScore

Example minimal fields for CSV exports

- EDC screening export: patient_id, site_id, date_screened, screen_status, failure_reason

- Enrollment export: patient_id, site_id, date_enrolled, randomized (0/1), date_randomized
- Visits export: patient_id, site_id, visit_name, scheduled_date, actual_date, visit_status, medication_adherence_pct, diary_submitted
- Site metadata: site_id, site_name, country, target_enrollment, PI_email

4) Data engineering & compliance (Week 1–2 tasks)

Goals: design model, ingest anonymized data, enforce compliance (PHI removal), and calculate base KPIs.

Week 1 (detailed)

Day 1–2: Study protocol & CRFs

- Read trial protocol and CRFs to determine which fields you need.
- Define what "adherence" means (visit completion, medication percentage, diary submissions).

Day 3: Data model design

- Draft star schema (use above schema).
- Create sample CSVs (fake/anonymized)
- **Day 4–7: Ingest & anonymize**
- Use Power Query to import CSVs.
- Remove / hash direct identifiers (name, address). Replace patient ID with pseudonym (PAT0001).
    - Example Power Query M to create a pseudonym:
    - Table.AddIndexColumn(Table.Distinct(#"PreviousStep"[patient_id]), "Index", 1, 1)
    - // then create "PAT" & Text.PadStart(Text.From([Index]),4,"0")
- Convert dates to proper date types, create Age or AgeBucket from DOB if allowed.

Compliance checklist

- Remove direct identifiers (names, contact), keep only necessary demographics.
- Log transformations for audit.
- Ensure dataset stored in secure location (e.g., encrypted workspace).

5) ETL transformations (Power Query & SQL patterns)

Common transforms:

- Normalize site codes (trim, uppercase).
- Create derived columns:
    - EnrollmentStatus = IF(RandomizedDate <> null, "Randomized", IF(EnrollmentDate <> null,"Enrolled","Screened")).
    - DaysFromScreenToEnroll = EnrollmentDate - ScreeningDate
    - VisitOnTimeFlag = ABS(actual_date - scheduled_date) <= allowed_window
- Aggregate per-site daily metrics:
    - EnrollmentsToday, ScreeningsToday, RandomizationsToday
- Rolling metrics: 7-day enrollment velocity per site.

Example Power Query M to calculate enrollment velocity (concept):

```
let
  Source = Csv.Document(File.Contents("enrollments.csv"),...),
  #"Changed Type" = Table.TransformColumnTypes(Source,{{"EnrollmentDate", type date}}),
  #"Grouped" = Table.Group(#"Changed Type", {"SiteID", "EnrollmentDate"}, {{"DailyEnrollments", each Table.RowCount(_), Int64.Type}}),
  #"AddedWeek" = Table.AddColumn(#"Grouped", "WeekStart", each Date.StartOfWeek([EnrollmentDate], Day.Monday)),
  #"WeeklyVelocity" = Table.Group(#"AddedWeek", {"SiteID","WeekStart"}, {{"EnrollmentsThisWeek", each List.Sum([DailyEnrollments]), Int64.Type}})
in
  #"WeeklyVelocity"
```

---

6) Power BI data model & relationships

- Load the dimension tables and fact tables into Power BI.
- Relationships: FactEnrollment[PatientPK] → DimPatient[PatientPK], FactVisits[SiteID] → DimSite[SiteID], etc.

- Mark date field to use a Date table (create a shared DimDate table for time intelligence).

---

7) Key DAX measures (copy/paste friendly)

Create measures for the dashboards.

Basic counts

```
Total Screened = COUNTROWS(FactScreeningEnrollment)

Total Enrolled = COUNTROWS(FactEnrollment)

Total Randomized = CALCULATE(COUNTROWS(FactEnrollment), FactEnrollment[RandomizedFlag] = 1)
```

Screen failure rate

```
Screen Failure Rate =

VAR total = [Total Screened]

VAR failures = CALCULATE(COUNTROWS(FactScreeningEnrollment), FactScreeningEnrollment[ScreenResult] = "Failed")

RETURN DIVIDE(failures, total, 0)
```

Enrollment velocity (per week)

```
Enrollments This Week =

CALCULATE(

  [Total Enrolled],

  DATESBETWEEN(DimDate[Date], STARTOFWEEK(TODAY(),1), TODAY())

)
```

Or better to compute rolling 7 days:

```
Enrollments (Last 7 days) =

CALCULATE([Total Enrolled], DATESINPERIOD(DimDate[Date], MAX(DimDate[Date]), -7, DAY))
```

Adherence % (per patient)

```
Patient Adherence % =

AVERAGE(FactVisits[MedicationTakenPercent])
```

Site Data Quality Score (example composite)

```
Site Data Quality =
VAR completeness = AVERAGE(FactDataQuality[DataCompletenessPct])
VAR timeliness = AVERAGE(FactDataQuality[TimelinessScore])
RETURN (completeness*0.6 + timeliness*0.4)
```

8) Dashboard pages & visuals (what to build)

Design for clarity — each page should answer specific manager questions.

1. **Enrollment Overview (main page)**
   - KPIs at top: Target vs Enrolled vs Remaining, %Complete, Forecasted Enrollment Completion Date
   - Time series chart: cumulative enrolled vs target line
   - Map/choropleth or site tiles with site progress
   - Date slicer and protocol arm filter

Forecast enrollment completion (simple approach)

- Use linear projection from recent velocity:
  - Remaining = TotalTarget - TotalEnrolled
  - Velocity = Enrollments (Last 14 days) / 14
  - DaysToComplete = Remaining / Velocity
  - ForecastDate = TODAY() + DaysToComplete
    Compute Velocity in Power Query or DAX and show forecast with caution/CI.

2. **Recruitment Funnel page**
   - Funnel visual: Screened → Eligible → Consented → Enrolled → Randomized
   - Funnel conversion rates and per-site mini-funnels
   - Table showing top bottleneck reasons (failure reasons)
3. **Site Performance Leaderboard**
   - Sortable table: SiteName, Enrolled, Randomized, Enrollment Velocity (per week), DataQualityScore, QueriesOpen
   - Conditional formatting for flags (underperforming sites)
4. **Patient Adherence Dashboard**

- Aggregate adherence trends over time (line chart)
- Distribution of adherence % (histogram)
- List of at-risk patients (with risk score and actionable fields: last visit date, missed visits count, adherence %)

5. Patient Detail (drill-through)

- Patient timeline (screening → enrollment → visits)
- Visit statuses & notes
- Predicted dropout risk and explanation

---

9) Dropout Risk Prediction (Week 3 tasks)

Goal: build a model to flag patients at high risk of dropout/non-adherence.

Data & features (suggested)

- Demographics: Age, Sex, Country/Region
- Site features: SiteEnrollmentVelocity, SiteDataQualityScore
- Behavior: CountMissedVisits, DaysSinceLastVisit, MedicationAdherencePct (rolling avg), eDiaryMissingCount
- Trial: Arm, NumberPriorAEs, ScreeningToEnrollDays
- Interaction features: missed visits × age group, adherence trend slope.

Label definition

- DroppedOut = 1 if patient discontinued (e.g., missed > X consecutive visits or status = discontinued) within Y days of enrollment. Pick threshold based on protocol (e.g., missed 2 consecutive visits).

ML workflow (concise)

1. Prepare dataset (one row per patient at a given cut-off time, include features up to that time).
2. Train/test split (time-aware if needed): use earliest N trials for training, latest for test OR use stratified split.
3. Handle imbalance (SMOTE or class weighting).
4. Models to try: Logistic Regression (baseline), Random Forest, XGBoost.
5. Evaluate: ROC AUC, precision-recall (esp. if positive class small), confusion matrix, calibration.

6. Explainability: SHAP or feature importances — show top drivers (missed visits, low adherence, long screening to enroll).
7. Export predictions as CSV and join back to patient table for Power BI.

Simple scoring pipeline (for classroom)

- Use scikit-learn LogisticRegression with class_weight='balanced'.
- Example steps: Impute missing, scale numeric features, encode categoricals (one-hot), train, persist model (pickle), score dataset, export predictions.

Where to score for reporting

- **Option A (recommended)**: Score offline (scheduled Python script / Azure ML job) and push scored CSV to storage read by Power BI. This keeps heavy compute off Power BI.
- **Option B**: For very simple logistic models, compute scores in Power Query using M or DAX (less flexible).

10) Row-Level Security (RLS) & access control (Week 4)

Implement RLS so site staff see only their site data; monitors see all.

Approach

- Create a UserSiteMap table with columns: UserPrincipalName (UPN), SiteID.
- In Power BI Desktop, create a role SiteUser with filter on DimSite:
- [SiteID] = LOOKUPVALUE(UserSiteMap[SiteID], UserSiteMap[UserPrincipalName], USERPRINCIPALNAME())
- Publish to Power BI Service and assign users to roles (or use Azure AD groups).

Notes

- If using email mapping, ensure USERPRINCIPALNAME() returns the user's email in service.
- Test RLS using "View as" in Desktop.

11) Testing, validation & regulatory review

- **Data validation tests**: row counts, expected ranges, null checks, reconciliation between raw and processed counts.

- **Metric validation**: cross-check enrollment counts by site/date against raw exports.
- **Reproducibility**: keep transformation scripts (Power Query steps documented).
- **Regulatory**: maintain an audit trail of ETL steps, data dictionary, and validation logs for audits.
- **User acceptance tests (UAT)**: trial manager confirms that KPIs match expectations and drill-throughs provide required details.

12) Deployment & schedule refresh

- Publish to Power BI Service into a secured workspace.
- Configure dataset credentials and gateway (if on-prem SQL).
- Set scheduled refresh frequency (daily or more frequent depending on needs).
- Document dataset refresh logs and error alerts.

---

13) Week-by-week granular plan (assignable to students)

Week 1 — Data engineering & modeling

- Day 1: Read protocol, list required fields.
- Day 2: Create sample anonymized CSVs.
- Day 3: Draft data model & relationships.
- Day 4–5: Power Query ingest and anonymization.
- Day 6: Create DimDate table & basic integrations.
- Day 7: Checkpoint deliverable: data model ER diagram + sample cleaned CSVs.

Week 2 — Recruitment funnel & site leaderboard

- Day 8: Build DAX measures for counts/ratios.
- Day 9: Build Enrollment Overview page (cumulative chart & forecast).
- Day 10: Build Recruitment Funnel page and bottleneck table.
- Day 11–12: Build Site Leaderboard and conditional formatting.
- Day 13–14: Mid-project review & QA session.

Week 3 — Adherence + risk model

- Day 15: Integrate visits & adherence logs.
- Day 16: Build adherence DAX measures and Adherence dashboard.
- Day 17–19: Prepare features & dataset for ML.
- Day 20–21: Train, evaluate dropout model and export scores; integrate scored file into Power BI and flag at-risk patients.

Week 4 — Security, polishing, deployment

- Day 22: Implement RLS & mapping table.
- Day 23: Performance optimization and refresh tuning.
- Day 24–25: Add summary & regulatory report page.
- Day 26: Create presentation & documentation.
- Day 27–28: Final review, user demo, QA signoff.

14) Deliverables & grading rubric

Deliverables

1. Cleaned anonymized dataset and ETL scripts (Power Query steps documented).
2. Power BI pbix with at least: Enrollment Overview, Recruitment Funnel, Site Leaderboard, Patient Adherence, Patient Detail pages.
3. DAX measure list (documented).
4. Dropout risk model code (Python notebook) + scored CSV and model performance report.
5. RLS proof & user mapping table.
6. Final slide deck: demo + validation evidence.

Focus how you develop -

- Data model & ETL correctness — 25%
- Visuals & UX clarity — 25%
- Correctness of KPIs & DAX measures — 15%
- ML model (reasonable approach, evaluation, integration) — 15%
- Security & compliance (anonymization, RLS) — 10%
- Documentation & presentation — 10%

15) Example quick checks / sample SQL queries

- Count enrollments by site:

```
SELECT site_id, COUNT(*) AS enrolled_count
FROM enrollments
GROUP BY site_id
ORDER BY enrolled_count DESC;
```

- Missed visit counts per patient:

```
SELECT patient_id, SUM(CASE WHEN visit_status='Missed' THEN 1 ELSE 0 END) AS missed_count
FROM visits
GROUP BY patient_id;
```

16) Tips

- **Ethics & compliance**: emphasize anonymization and minimal data principle.
- **Forecasting caution**: show managers confidence intervals and warn about extrapolating from limited data.
- **Explainability**: always show why a patient is flagged (top features) — managers need actionable reasons.
- **Data quality**: poor site data quality will ruin the model; include data quality as a feature and teach students reconciliation.
- **Real-time**: true real-time requires streaming; this project can be "near real-time" with frequent scheduled refreshes.

17) Optional advanced extensions (extra credit)

- Integrate SMS/email alert automation when patient risk score > threshold.
- Add survival analysis to estimate time-to-dropout.
- Use SHAP visualizations inside Power BI (precompute SHAP values and import).
- Create mobile-friendly report view for site monitors.

18) Quick starter checklist

1. Create anonymized sample CSVs for screening, enrollment, visits, site metadata.
2. Build Power Query transforms to clean + pseudonymize.
3. Construct star schema in Power BI.
4. Implement DAX measures for core KPIs.
5. Build Enrollment Overview and Recruitment Funnel visuals.
6. Train & export dropout model (simple logistic baseline).
7. Import scored results and add an "At Risk" visual.
8. Set up RLS and validate with sample user accounts.
9. Finalize report page + documentation.