“**Build with Us | Speedrun: Data Science Fundamentals**”是 learn.palantir.com 上的一门实战课程，旨在指导用户如何结合使用 **Foundry** 的代码与无代码工具，完成从原始数据处理到交互式成果发布的完整数据科学工作流 1, 2。
以下是根据来源对该课程内容的详细解释：

1. 核心目标与业务场景

- 核心目标：该课程强调“广度优先”，带用户快速走通从非结构化数据（PDF）和结构化数据（CSV）中提取、分析并可视化见解的全过程 1, 2。
- 业务背景：课程模拟了一个临床研究（**Clinical Study**）场景，研究药物对患者产生的不良反应（**Adverse Events**）3, 4。
- 原始数据：包含患者的人口统计数据（CSV）、不良反应记录（CSV）以及患者反馈调查（PDF 扫描件）5-7。

2. 核心技术环节：端到端工作流

该课程将工作流分为三个主要阶段：

第一阶段：使用 Pipeline Builder 进行数据预处理与 AI 集成

- 非结构化数据处理：利用 Pipeline Builder 将 PDF 转化为文本，并使用 join array 将多页内容合并为单一字符串 8-10。
- **AIP** 赋能：集成大语言模型（如 **Gemini** 或 **GPT-4**），通过编写提示词（Prompt）自动从患者反馈中提取“患者 ID”、“情感评分”和“文字摘要” 11-14。
- 数据整合：将清洗后的 CSV 列表与 AI 提取的结构化数据进行左连接（**Left Join**），生成一张包含人口统计学特征、用药分组合反馈摘要的汇总表 15-17。

第二阶段：使用 Jupyter Workspace 进行深度分析

- 环境准备：在 Foundry 内部启动 **Jupyter Lab**，并安装 **Matplotlib** 和 **Pandas** 等 Python 库 18-21。
- 见解生成：编写 Python 代码分析数据，例如构建直方图来观察不同年龄段患者在“安慰剂组”与“治疗组”中的分布，以及不同不良反应类别的年龄分布趋势 22, 23。

第三阶段：使用 Streamlit 发布交互式报告

- 应用构建：使用 **Streamlit** 框架将 Python 脚本转化为 Web 应用程序 24, 25。
- 交互功能：构建一个“不良反应年龄分布探索器（**Adverse Event Age Distribution Explorer**）”，允许最终用户通过下拉菜单选择特定症状（如焦虑、食欲下降），实时查看该群体的年龄分布图和患者反馈摘要 26-29。

3. 关键优势：维护数据血缘（Data Lineage）

课程强调在 Foundry 中进行数据科学工作的独特价值在于维护数据出处（**Data Provenance**）30。

- 可视化追踪：通过 **Data Lineage** 应用，用户可以清晰地看到从原始 PDF 和 CSV 到最终 Streamlit 应用的完整链路 30, 31。
- 可重现性：即使是无代码的 Pipeline Builder 步骤，也会生成**伪代码（Pseudo-code）**供分析师审计，确保研究结果的透明和可追溯 31, 32。

4. 课程总结

该课程为数据科学家提供了一套完整的工具箱，展示了如何不再局限于本地机器的孤岛式开发，而是利用 **Pipeline Builder**、**AIP Logic** 和 **Jupyter** 构建起一套工业级、受治理且可自动更新的临床分析系统 2, 30。