OpenEDG™ Python Institute
Certifying Python Skills for Real Careers
https://pythoninstitute.org

PCAD™ – Certified Associate Data Analyst with Python 試験シラバス

（試験コード：PCAD-31-02）

最終更新日：2025年7月14日

1. データ取得と前処理（14）

1.1 データの収集・統合・保存（3）

1.1.1 データ収集方法の説明と研究・ビジネス・分析における活用の比較

A. アンケート、インタビュー、Webスクレイピングなど、さまざまな手法の検討
B. 代表性のあるサンプリング、データ収集時の課題、定性調査と定量調査の違いについて議論
C. データ収集における法的・倫理的な観点の検討
D. 特に個人を特定できる情報（PII）に関して、プライバシーと機密性の維持におけるデータ匿名化の重要性の説明
E. データ収集が、ビジネス戦略の策定・市場調査の精度・リスク評価・政策立案・ビジネス上の意思決定に与える影響の調査
F. 調査設定、対象者の選定、構造化インタビューなどを含む、データ収集プロセスと手法の説明

1.1.2 複数ソースからのデータ集約およびデータセットとしての統合

A. データベース、API、ファイルベースのストレージなど、さまざまなソースからデータを統合する手法の説明
B. データ形式の違いや整合性の問題など、データ収集時に生じる課題への対処
C. 統合されたデータセットにおけるデータの一貫性と正確性の重要性の理解

1.1.3 さまざまなデータ保存ソリューションの説明

A. 複数のデータ保存方法と、それぞれが適している利用シーンの理解
B. データウェアハウス、データレイク、CSVやExcelなどのファイルベースの保存方式の概念の区別
C. クラウドストレージソリューションの概念と、データマネジメントにおける役割の拡大について説明

1.2 データクレンジングと標準化（4）

1.2.1 構造化データと非構造化データの理解およびデータ分析における違いと影響の把握

A. データベースやスプレッドシートのような構造化データの特徴と、分析における扱いやすさについて把握
B. テキスト、画像、動画などの非構造化データに関する理解と、分析における追加処理の必要性について理解
C. データ構造が、データの保存・取得・分析手法にどのような影響を与えるかの検討

1.2.2 誤ったデータの特定、修正、または除去

A. 各種診断手法を用いた、データの誤りや不整合の特定
B. 欠損、不正確、誤解を招く情報への対処
C. 数値データの問題、重複レコード、不正なデータ入力、欠損値など、特定のデータ品質の問題に取り組む
D. 欠損が完全にランダム（MCAR）、条件付きランダム（MAR）、ランダムではない（MNAR）の違いの説明と、それぞれが分析に与える影響の理解
E. 各種データ補完手法を含め、欠損データへの対処方法について検討
F. データを修正または除去することが、全体のデータ完全性および分析結果に与える影響を理解
G. 外れ値検出の文脈におけるデータ収集の重要性の説明
H. 高品質なデータが、正確な外れ値検出にとって重要な理由の説明
I. データ型（数値、カテゴリなど）の違いが、外れ値検出の戦略にどのように影響するかの説明

1.2.3 データ正規化とスケーリングの理解

A. 異なる変数を比較可能な同程度のスケールに揃えるために、データ正規化が必要となる理由の理解
B. Min-Max スケーリングやZスコアによる標準化など、代表的なスケーリング手法の理解
C. One-hot エンコーディングやラベルエンコーディングなど、カテゴリ変数を定量分析向けに符号化する方法の説明
D. データ削減によって検討対象となる変数の数を減らしたり、モデルを単純化することの利点と、データ説明の可能性が損なわれることによる欠点の説明
E. 外れ値検出および処理など、データ品質を確保するための外れ値対応手法の説明
F. 特に日付時刻形式や数値表現を扱う際に、異なるデータセット間でのデータ形式標準化の重要性について理解

1.2.4 データクレンジングおよび標準化手法の適用

A. データ補完、文字列操作、データ形式の標準化、ブール値の正規化、文字列の大文字・小文字正規化、文字列から数値への変換などの実行
B. 補完と除外のメリット・デメリット、およびそれらが分析の信頼性と妥当性に与える影響の議論
C. One-hot エンコーディングの概念を説明し、カテゴリ変数を二値形式に変換して機械学習アルゴリズムに適した形にする方法の説明
D. 連続変数をカテゴリ変数へ変換する際に用いられるビニング（バケタイズ）という概念と、その応用の説明

1.3 データ検証とデータ完全性（2）

1.3.1 基本的なデータ検証手法の実行と理解

A. 型チェック、範囲チェック、項目間チェックといった「検証」の種類の定義と、それぞれのツール（Pythonロジックやスキーマチェックなど）への対応付け
B. 型チェック、範囲チェック、クロスリファレンスチェックの実行
C. データ取り込みスクリプトの早い段階で型チェックを行うことの利点について説明

1.3.2 明確な検証ルールによるデータ完全性の確立と維持

A. データ完全性の概念および信頼性が高く正確なデータベースを維持する上での重要性の理解
B. データの正確性と一貫性を確保する明確な検証ルールの適用

1.4 データ前処理手法（5）

1.4.1 データ取得におけるファイル形式の理解

A. CSV（表形式データ）、JSON（構造化データ）、XML（階層構造データ）、TXT（非構造化テキスト）など、代表的なデータファイル形式の役割と特徴の説明
B. これらのファイル形式をデータ分析ツールでインポート・エクスポートする基本的な方法を理解し、実務での活用に焦点を当てる

1.4.2 データセットへのアクセスと管理、および有効活用

A. ローカルファイル、データベース、オンラインリポジトリなど、さまざまなソースからデータセットへアクセスする基本の理解
B. 分析準備の一環として、データの整理、ソート、フィルタリングなどのデータ管理の基本原則について理解

1.4.3 さまざまなソースからのデータ抽出

A. データベース、API、オンラインサービスなどからデータを取得・集約するための基礎的なデータ抽出方法の説明

B. Pythonのツールやライブラリ（BeautifulSoup、requests）を用いたHTMLからのデータ抽出
C. データ互換性や完全性など、データ抽出時の基本的な課題や留意点の理解
D. Robots.txt の遵守やリクエストのレート制限への配慮など、倫理的なWebスクレイピングの実践について議論

1.4.4 スプレッドシートの可読性とフォーマットのベストプラクティスの適用

A. レイアウト調整、フォーマットの工夫、基本的な数式の利用などを通じた、スプレッドシート内のデータの可読性と利便性の向上

1.4.5 データ分析のための準備・調整・前処理

A. 周囲の文脈、目的、ステークホルダーの期待を理解し、それに基づくデータ準備のステップを設計する重要性の理解
B. ソート、フィルタリングなど、分析作業に向けたデータセット準備を含む前処理の基本概念の理解
C. 日付時刻形式の統一やデータ構造の整合など、分析に適した適切なデータ整形の重要性の議論
D. ワイド形式（wide format）とロング形式（long format）など、分析に適した形式へデータを変換するためのデータセット構造化の基本導入
E. 特に機械学習プロジェクトにおいて、モデル検証のためにデータを訓練データテストデータに分割するという概念と、その重要性について説明
F. 前処理段階における外れ値対応がデータ品質に与える影響について理解

2. プログラミングとデータベースのスキル（16）

2.1 Pythonの基礎スキル（5）

2.1.1 データ関連の問題を解決するためのPython構文と制御構造の適用

A. 変数、スコープ、データ型など、Pythonの基本構文の正しい使用
B. ループや条件分岐などの制御構造を用いた、データフローの適切な制御

2.1.2 Pythonの関数の分析と設計

A. 目的が明確な関数の設計と、位置引数やキーワード引数の適切な使用
B. 必須引数とオプション引数の違いの理解と、状況に応じた効果的な使い分け

2.1.3 Pythonにおけるデータサイエンスのエコシステムの評価と活用

A. データサイエンスで多用される主要なPythonライブラリやツールの特定
B. さまざまなデータ分析の場面における、適切なリソース（ライブラリやツール）の批判的な評価

2.1.4 Pythonの基本データ構造を用いたデータ操作

A. タプル、集合、リスト、辞書、文字列などを用いたデータの整理と操作
B. データ処理の課題に応じた、最適なデータ構造の選択と複雑なデータの操作

2.1.5 Pythonスクリプトにおけるベストプラクティスの理解と実践

A. PEP 8 に基づく記述の実践
B. PEP 257 の理解と、可読性と保守性の高いコードの説明文（dogstring）の作成

2.2 モジュール管理と例外処理（2）

2.2.1 モジュールのインポートと PIP によるパッケージ管理

A. 標準インポート、選択的インポート、エイリアス（別名）インポートなどの使い分け
B. 標準ライブラリ、PIP 経由の外部パッケージ、ローカルで作成したモジュールなど、さまざまなソースからモジュールを読み込む方法の理解
C. タスクに応じた必要なモジュールの特定と、それぞれの目的や機能の理解
D. PIP を用いたパッケージのインストール・更新・アンインストールの実行

2.2.2 基本的な例外処理によるスクリプトの保守性向上

A. try-except 文などを用いたエラーの補足・処理をする基本的な例外処理の仕組みの実装
B. よくあるエラーの予測と、効果的な対処法の確立
C. エラーメッセージの読み取りおよび問題の原因の特定・修正による、より堅牢なスクリプトの作成

2.3 データモデリングのためのオブジェクト指向プログラミング（3）

2.3.1 オブジェクト指向の基礎を用いたデータの構造化

A. コンストラクタやインスタンス変数を含むクラスの定義およびデータ構造の表現
B. コンストラクタやインスタンスメソッドを用いた属性と振る舞いの整理
C. カプセル化の原則の適用および命令規則（例：_protectedや__private）とメソッドによるアクセス（ゲッター/セッター）を用いた、オブジェクト内部状態の適切な管理

2.3.2 コードの再利用と可読性を高めるためのオブジェクト指向デザイン

A. コンポジションを用いた関連データモデルの集約（例：User オブジェクトをResponse オブジェクトに含める）

B. 継承を用いた基底クラスの拡張と、メソッドをオーバーライドした特定の振る舞いの実現（例：複数のエクスポートクラス）
C. 異なるサブクラスで同じメソッド（例：process()、export()）を呼び出すことによるポリモーフィズムの提示

2.3.3 データ処理パイプラインにおけるオブジェクトの同一性と比較の管理

A. 参照を保持する変数を用いて、オブジェクトが共有されている場合と独立している場合の挙動（例：オブジェクト内リストの変更）の違いについて理解
B. == と is の使い分けによる内容の等価性の定義（必要に応じて __eq__() を実装）

2.4 データ分析のための SQL（6）

2.4.1 SQL によるデータの取得と操作

A. SQLクエリの作成・実行およびデータベースからのデータ抽出
B. SQL関数や句を用いた、データの加工・絞り込み
C. SELECT、FROM、JOIN（INNER・LEFT・RIGHT・FULL）、WHERE、GROUP BY、HAVING、ORDER BY、LIMIT などを組み合わせた SQL クエリの構築
D. データの取得要件の分析と、必要に応じた各句（SFJWGHOL）の組み合わせ

2.4.2 SQL による基本的な CRUD 操作（データの作成・取得・更新・削除）

A. CREATE、READ、UPDATE、DELETE 文の正しい操作
B. データの挿入、取得、更新、削除を行う SQL 文の構築

2.4.3 Pythonからデータベースへ接続

A. sqlite3 や pymysql などのライブラリを用いた、Python からデータベースに接続する方法の理解と実装
B. 接続時に発生する一般的な問題の分析と適切な対処

2.4.4 Pythonでパラメータ化された SQL クエリの実行

A. パラメータを用いた SQL クエリを Python から実行することによる、安全なデータベースの操作
B. パラメータ化クエリを使うことで、SQL インジェクションを防ぎ、データ整合性を保つ利点の評価

2.4.5 SQL のデータ型の理解および Python スクリプト内での適切な取り扱い

A. SQL の主要なデータ型と、それに対応する Python のデータ型の理解
B. SQL と Python 間でデータをやり取りする際の、適切な型変換

2.4.6 データベースの基本的なセキュリティとSQLインジェクション対策の理解

A. SQL インジェクションを含む、データベースセキュリティの基本原則について理解
B. Python 環境で安全な SQL クエリを書くための実践的な方法の適用

3. 統計解析（4）

3.1 記述統計（2）

3.1.1 データ分析における統計的な指標の理解と適用

A. 中心傾向と散らばりの指標の理解と説明
B. 基本的な統計分布（正規分布、一様分布）を把握し、時系列データ・単変量・二変量・多変量といったさまざまな文脈における傾向の読み取り
C. 信頼区間などの信頼性指標を計算に用いた、データの信頼性の評価

3.1.2 データ間の関係性の分析と評価

A. データセットの分析による外れ値の検出と、ピアソンの相関係数を用いた正の相関・負の相関の強さに関する評価
B. 箱ひげ図、ヒストグラム、散布図、折れ線グラフ、相関ヒートマップなど、さまざまなグラフからの情報読み取りと、その妥当性の批判的な評価

3.2 推測統計（2）

3.2.1 Bootstrap による標本分布の理解と活用

A. Bootstrap 法の理論的背景と統計的な考え方の理解
B. Bootstrap の文脈における、離散データと連続データの違いと扱い方の理解
C. Bootstrap が標本分布の推定手法として有効な状況や、データの種類の見極め
D. Python を用いた Bootstrap サンプリングの実装および標本分布の生成と分析
E. Bootstrap で得られた結果に関する信頼性や妥当性の多角的かつ統計的な状況の評価

3.2.2 線形回帰とロジスティック回帰の使いどころと限界の説明

A. 線形回帰の理論、前提条件、数学的な枠組みの理解
B. ロジスティック回帰の考え方、典型的な利用場面、統計的な背景の説明
C. データの性質や分析目的（予測したい値・目的変数の種類）に応じた、線形回帰とロジスティック回帰の適切な選択
D. 離散データ・連続データの概念を適用した、線形回帰・ロジスティック回帰モデルの選択および実装
E. Python を用いて線形回帰・ロジスティック回帰モデルを適用し、パラメータ推定とモデルフィッティングを行う方法の提示
F. 回帰分析の結果（係数や決定係数などのモデル適合度指標）の正しい解釈
G. 線形回帰・ロジスティック回帰モデルの前提条件、限界、バイアスの可能性の把握と、それが結果に与える影響について説明

4. データ分析とモデリング（9）

4.1 Pandas と NumPy によるデータ分析（6）

4.1.1 Pandasを用いたデータの整理とクレンジング

A. Pandasを用いた、表形式データのフィルタリング、ソート、欠損値や不整合データの処理
B. 基本的なデータクレンジング手法を適用した、生データの分析可能な状態へのフォーマット

4.1.2 Pandasによるデータ結合と再構造化

A. merge、join、ピボット、リシェイプなどの高度なデータ操作による、複数のデータフレームの結合・変形
B. 目的の分析ワークフローに適した形へのデータセット構造の設計および整形

4.1.3 Series と DataFrameの関係性の理解

A. Pandas の Series と DataFrame の概念的な違いと関係性の説明
B. インデックス操作やベクトル化された関数を用いた、データの効率的な参照および変換

4.1.4 ロケータとスライシングによるデータアクセス

A. Loc、iloc、スライシング、条件抽出を用いた、必要なデータの正確な取得および更新
B. 適切なインデックス設計と指定方法による、効率的で正確なデータアクセスの実現

4.1.5 配列演算と基本データ構造の使い分け

A. NumPy を用いた、配列に対する四則演算、ブロードキャスト、集約処理などの演算処理
B. 配列（NumPy配列）、リスト、Series、DataFrame、ndarray などのデータ構造の特徴と用途・性能面での違いの理解および使い分け

4.1.6 データの集計・要約と洞察の抽出

A. groupby() によるグルーピング、ピボットテーブルやクロス集計（クロスタブ）を用いた、要約表の作成
B. Pandas や NumPy の記述統計機能を用いた、トレンドの把握や異常値の検知および意思決定の支援につながる統計量の算出

4.2 統計手法と機械学習（3）

4.2.1 Python による記述統計を用いたデータセット分析

A. Pythonによる平均、中央値、最頻値、分散、標準偏差などの主要な統計量の計算およびその意味の解釈
B. Pandasや NumPy などのライブラリを用いた、実データに対する記述統計の算出および分析

4.2.2 モデル評価におけるテストデータの重要性の理解

A. 機械学習モデルの性能検証における、テストデータが果たす役割の理解
B. バイアスのない正確な評価を行うための、テストデータの適切な分割・選定および利用方法の説明

4.2.3 教師あり学習アルゴリズムとモデル精度の分析と評価

A. 代表的な教師あり学習アルゴリズムの特徴や適用ケースの整理と説明
B. 過学習と過少適合、およびバイアスとバリアンスのトレードオフの概念の理解と説明
C. 線形回帰およびロジスティック回帰が、このトレードオフにおいて示す傾向の評価と、その理解を活かしたモデル精度の問題を防ぐ方法の考察

5. データの伝達と可視化（5）

5.1 データ可視化の手法（3）

5.1.1 Matplotlib と Seaborn による基本的な可視化スキルの習得

A. Matplotlib と Seaborn を用いた、箱ひげ図、ヒストグラム、散布図、折れ線グラフ、相関ヒートマップなど、さまざまな種類のグラフの作成
B. これらの可視化グラフに表現されたデータや傾向の読み取りと、その洞察を得た結果の明瞭な伝達

5.1.2 表現方法ごとの長所・短所の評価

A. データの種類や分析の目的に応じた、適切なグラフタイプの評価
B. 選択した可視化が意図したメッセージや洞察を適切に伝えられているかに関する批判的な分析

5.1.3 明確に伝わるようにグラフを調整・注釈するスキルの育成

A. グラフにタイトル、軸ラベル、注釈を付与することによる視認性の向上
B. 可視化を利用した探索的データ分析を通じた仮説の組み立ておよびデータに基づく洞察の検証
C. 可視化結果をもとに、データに基づく意思決定の練習
D. 散布図などのプロットで色をカスタマイズすることによる、可読性と識別性の向上
E. 軸ラベルやタイトルの適切な設定と、データの意味の直感的な伝達

F. 凡例の位置、フォントサイズ、背景色などのプロパティの調整による、識別性と可読性の改善

5.2 データから得られた洞察の効果的な伝達（2）

5.2.1 相手に合わせたコミュニケーションおよび可視化とテキストの組み合わせ

A. 対象者の背景・関心・知識レベルの分析およびそれに応じた伝え方の検討
B. 多様な対象（技術者・非技術者など）のニーズや期待に合わせた、説明のスタイルや内容の調整
C. 技術系・非技術系の両方のステークホルダーに向けた、データから得られた洞察を分かりやすく伝えるプレゼンテーションやレポートの作成
D. グラフやチャートをケースの流れに沿う形で資料に組み込むことによる、ケース全体との一貫性の維持
E. 簡潔で情報量のあるテキストを用いた、データ可視化・文脈・重要なポイント（キーメッセージ）の補足
F. ビジュアルとテキストの要素が相互補完し、データの理解を高めるよう構成
G. スライドの情報量を適切に抑え、不要な要素を減らすことによる主要なメッセージへの集中
H. データから得られた洞察を軸に、行動につながるストーリーとしての「データストーリーテリング」の組み立て
I. 明確さとアクセシビリティを意識した一貫したカラーパレットの選定と、資料全体にわたる統一的な使用

5.2.2 主要な知見の要約およびエビデンスと論理的理由による根拠づけ

A. データ分析による重要な知見の抽出およびそのプロセスの理解
B. 複雑な情報を、簡潔で意味のある要約として整理する手法の習得
C. 文脈に応じた、最も関連性の高い洞察の優先的な協調
D. 主張や結論は、必ずデータに基づく根拠と論理的な説明で支えることの重要性に関する理解
E. なぜその結論や推奨が導かれたのか、根拠となるデータと推論プロセスの明確な説明
F. 主張や提案を支えるエビデンスを、利き手にとって分かりやすい形で提示できるスキルの習得

MQC プロファイル

PCAD™ - Certified Associate Data Analyst with Python の MQC（最小合格候補）は、実務での初級レベルのデータ分析業務を支援できるだけの、必要最低限かつ本質的なスキルと知識を身につけていることが求められます。

受験者は、Python・SQL・一般的なデータ分析ツール/ライブラリを用いて、データの取得、クレンジング、前処理、分析、モデリング、および結果の共有・報告までの一連の流れを理解している必要があります。また、データベース、スプレッドシート、API、HTMLのWebページなど、さまざまなデータソー

スに接続し、requests や BeautifulSoup などのツール/ライブラリを用いて、必要なデータを取得できなければなりません。

MQCは、変数、関数、制御構造、リスト・辞書・セットなどのデータ構造を用いた Python スクリプトを、読みやすく整理して記述することができます。ドキュメンテーション、エラー処理、モジュール化などの Python のベストプラクティスを実践し、pip を用いたパッケージ管理も行うことができます。

さらに、Pandas、NumPy、statistics などのライブラリを用いて、構造化データのクレンジング、整形、分析を行い、記述統計量、相関、基本的な集計指標を計算できます。SQL を用いてデータの取得・操作を行い、sqlite3 を用いて Python スクリプトからデータベースに接続することも求められます。また、パラメータ化クエリを用いてデータの整合性を保ち、SQL インジェクションを防ぐ方法を理解している必要があります。

候補者は、線形回帰やロジスティック回帰などの基本的な統計モデルを扱い、ブートストラップなどの推測統計手法を適用できます。また、モデルの検証、テストデータによる分割評価の重要性、および過学習リスクについても理解していることが期待されます。

最後に、MQC は Matplotlib や Seaborn を用いて、わかりやすく洞察に富んだ可視化を作成し、受け手に応じた「データストーリー」を構成できる必要があります。設計やコミュニケーションレベルのベストプラクティスに基づき、書面および口頭の両方の形式で、分析結果を効果的に伝えられることが求められます。

ブロック1：データ取得と前処理

配点比率：全体の29.2%（14問）

出題範囲：
MQCは、データとは何か、どのように構造化されるか、どのような手順で分析可能な情報に変換されるかを理解している必要があります。データ型として、構造化データ、半構造化データ、非構造化データの違いを説明でき、それぞれが保存方法、処理方法、分析手法にどのような影響を与えるかの説明も求められます。また、アンケートやインタビュー、API、BeautifulSoup などのツールを用いたWebスクレイピングといった、さまざまなデータ収集手法を説明し、それらが調査・ビジネス・分析の現場でどのように使われるかを理解していることが前提となります。

さらに、CSV、JSON、Excel、データベース、データレイク、データウェアハウスなど、データ型に応じた適切な保存形式やストレージを選択でき、クラウドストレージが現代のデータエコシステムで果たす役割を説明できる必要があります。不適切なデータ収集や保存の実践が、後続の分析プロセスでデータ品質の低下やエラーの原因になることも理解していることが求められます。

MQCは、複数のソースからデータを統合し、集約の際に発生する不整合を解消することができます。フォーマットの揃え方、型の不一致、スキーマの違いがもたらす影響を理解しており、欠損値・重複データ・不正な値の特定の修正など、基本的なデータクレンジング手法を適用できます。また、カテゴリ変数のエンコーディング、数値データのスケーリング、日付・時刻データの形式統一といった処理の重要性も理解しています。

型チェック、範囲チェック、クロスチェック（クロスリファレンス）のような基本的な検証手法を用いて、データ品質と整合性を確保できることが求められます。さらに、ソート、フィルタリング、ワイド形式・ロング形式へのリシェイプ、モデル構築に備えた訓練データとテストデータへの分割などを通して、分析に適した形にデータを整えることができます。また、個人データを扱う際の倫理的・法的責任（匿名化、同意取得、GDPR や HIPAA などの枠組みへの準拠）についても理解している必要があります。

ブロック2：プログラミングとデータベースのスキル

配点比率：全体の33.3%（16問）

出題範囲：
MQCは、Python を用いてデータ処理タスクを行うための十分なプログラミングスキルを備えている必要があります。整数・浮動小数点数・文字列・ブール値といった基本データ型や、リスト・辞書・タプル・セットなどの代表的なデータ構造を使って、変数の定義・更新・操作が行えます。また、パラメータと戻り値を持つ関数を用いて、再利用性の高い分かりやすいコードを書き、条件分岐やループなどの制御構造を使って、データの処理・分析を効率的に進められることが求められます。

クリーンコードの考え方を理解しており、インデントや命令規則、PEP 8・PEP 257 に沿ったドキュメンテーションなどのベストプラクティスを意識してコードを書きます。また、csv、math、statistics、datetime、collections などの標準ライブラリに親しみがあり、pip を使って外部パッケージのインストール・管理ができることも求められます。

MQCは、オブジェクト指向プログラミング（OOP）の基本を理解し、データを構造化して扱うために活用できます。クラスを定義し、オブジェクトを生成し、その中に属性やメソッドを整理して配置するきおとで、再利用性が高く見通しのよいデータ処理ワークフローを構築できなければなりません。

Python に加えて、構造化データの取得と操作のために SQL を使いこなすことも期待されています。SELECT、WHERE、各種 JOIN 句を用いてデータを取得・絞り込み、GROUP BY、HAVING、ORDER BY などを使って集計やグルーピングを行えます。また、INSERT、UPDATE、DELETE といった CRUD 操作を行う SQL 文を記述できる必要があります。

さらに、sqlite3 などのライブラリを利用して、Pythonスクリプトからリレーショナルデータベースに接続し、パラメータ化クエリを実行して SQL インジェクションからシステムを保護しつつ、データの整合性を保てることが求められます。加えて、データの取得・挿入時に、SQL側とPython側のデータ型を適切に変換する方法を理解していることも重要です。

ブロック3：統計解析

配点比率：全体の8.3%（4問）

出題範囲：
MQCは、統計の基礎概念をしっかり理解し、記述統計を用いてデータセットを要約できることが求められます。平均・中央値・最頻値といった代表値や、分散・標準偏差といったばらつきの指標を理解し計算でき、正規分布や一様分布など基本的な分布の種類を説明できなければなりません。

またピアソンの相関係数を用いて変数間の関係性を評価し、外れ値を視覚的・統計的に特定できる必要があります。ヒストグラム、箱ひげ図、散布図といった可視化が、データの分布や傾向を探索的に理解するための重要な手段であることも理解しています。

推測統計の観点では、理論分布がはっきりしない場合に標本分布を推定する手法として、ブートストラップに慣れていることが求められます。離散データと連続データの違いを理解し、信頼性評価などでブートストラップが適切に使える場面を判断できなければなりません。

さらに、線形回帰とロジスティック回帰を説明・適用でき、その前提条件（仮定）を理解したうえで、回帰係数やモデル適合度指標などの出力を解釈できることが求められます。過学習といった典型的な限界にも注意を払い、モデル検証の重要性について説明できることが、MQCに期待される能力です。

ブロック4：データ分析とモデリング

配点比率：全体の18.8%（9問）

出題範囲：
MQCは、Pandas と NumPy を用いたデータ分析に十分な習熟度を持っていることが求められます。Pandas では、dropna()、fillna()、sort_values()、replace() などの関数を使ってデータのクレンジングと整理ができ、pivot()、melt()、groupby()、merge() といったメソッドでデータの形を変えたり再構造化したりできます。さらに、DataFrame と Series の違いと役割を理解し、.loc や .iloc を用いてデータに正確にアクセス・更新できる必要があります。

NumPyでは、要素ごとの計算、集約処理、配列のブロードキャストなどの数値演算を行うことができ、大規模データに対してPythonの標準リストよりもNumPy配列が高いパフォーマンスを発揮する理由を理解している必要があります。記述統計による要約や、ビニング（区間分割）、スケーリング、エンコーディングといった基本的な特徴量エンジニアリングを行い、モデリングの準備を整えることも求められます。また、グループ集計や条件付きフィルタを組み合わせてPandasとNumPyを活用し、データセットから有用な洞察を抽出できなければなりません。

さらに、教師あり学習の基本的な流れを理解し、train/test split によってモデルを評価する手順を身につけていることが重要です。基本的な線形回帰・ロジスティック回帰モデルを当てはめて結果を解釈でき、精度（accuracy）、過学習（overfitting）、過小適合（underfitting）、バイアスとバリアンスのトレードオフといった主要なモデリング概念について理解していることが、MQCに期待される内容です。

ブロック5：データの伝達と可視化

配点比率：全体の10.4%（5問）

出題範囲：
MQCは、Matplotlib と Seaborn を使って、データを効果的に可視化・解釈できることが求められます。データの種類や目的に応じて適切なグラフを選び、棒グラフ、ヒストグラム、散布図、箱ひげ図、折れ線グラフ、相関ヒートマップなどを用いて、傾向やパターンを分かりやすく表現できます。

また、ラベル、タイトル、凡例、グリッド線、カラースキームなどを追加してグラフの可読性を高める方法を理解しています。注釈を加えて重要なポイントを強調し、見た目を調整して、情報が伝わりやすいチャートに仕上げることができます。

MQCは、簡潔で分かりやすい要約テキストと可視化を組み合わせることで、分析結果をさまざまな相手に伝えられます。データチームのような技術的なステークホルダーにも、マネージャーやクライアントのような非技術系の受け手にも、それぞれに合わせてメッセージや説明の深さを調整することができます。

さらに、ビジネスや研究上の問いと分析結果を結びつける「データストーリー」を組み立て、結論や提言がデータに裏打ちされていることを示せる必要があります。スライド資料やレポートのデザイン原則を理解し、情報過多やあいまいな表現、誤解を招くグラフ表現を避ける、といった点にも配慮できることが期待されています。

合格要件

PCAD試験に合格するためには、すべての出題セクションにおける平均正答率が**75%**以上であることが求められます。

PCAD-31-02 試験構成概要

PCAD™試験は、PythonとSQLを活用してデータの取得～加工・分析・モデリング・教諭を行う能力を評価する、単一選択式および複数選択式の設問で構成されています。各問題の配点は形式に関わらず最大1点として採点されます。試験終了後、総得点は正規化され、受験者の成績はパーセンテージで報告されます。

試験は5つのブロックから成り、それぞれデータ分析やプログラミング、SQLクエリの技術的なコアスキルの領域をカバーします。各分野の問題数および内容の難易度に応じて比例的に配点が設定されています。

ブロック番号	ブロック名	出題数	配点
1	データ取得と前処理	14	29.2%
2	プログラミングとデータベースのスキル	16	33.3%
3	統計解析	4	8.3%
4	データの分析とモデリング	9	18.8%
5	データの伝達と可視化	5	10.4%
	合計	**48**	**100%**