

No	大分類	小分類	設問主旨	難易度	設問内容
P-001	列に対する操作	全項目指定	全項目を指定行数抽出する	★	レシート明細データ (df_receipt) から全項目の先頭10件を表示し、どのようなデータを保有しているか目視で確認せよ。
P-002	列に対する操作	列指定	特定の列を抽出する	★	レシート明細データ (df_receipt) から売上年月日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、10件表示せよ。
P-003	列に対する操作	列名変更	指定列の列名を変更する	★	レシート明細データ (df_receipt) から売上年月日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、10件表示せよ。ただし、sales_ymdsales_dateに項目名を変更しながら抽出すること。
P-004	行に対する操作	単一条件	特定条件に合致する行を抽出(=,>,<)	★	レシート明細データ (df_receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001"
P-005	行に対する操作	複数条件	複数条件に合致する行を抽出する①	★	レシート明細データ (df_receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 売上金額 (amount) が1,000以上
P-006	行に対する操作	複数条件	複数条件に合致する行を抽出する②	★	レシート明細データ (df_receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上数量 (quantity)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 売上金額 (amount) が1,000以上または売上数量 (quantity) が5以上
P-007	行に対する操作	範囲指定	複数条件に合致する行を抽出する③	★	レシート明細データ (df_receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 売上金額 (amount) が1,000以上2,000以下
P-008	行に対する操作	不一致	特定条件に合致しない行を抽出する (!=)	★	レシート明細データ (df_receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 商品コード (product_cd) が"P071401019"以外
P-009	行に対する操作	補集合	ド・モルガンの法則で書き換える	★	以下の処理において、出力結果を変えずにORをANDに書き換えよ。  df_store.query('not(prefecture_cd == "13"   floor_area > 900)')
P-010	あいまい条件	前方一致	データの前方一致で条件指定する	★	店舗データ (df_store) から、店舗コード (store_cd) が"S14"で始まるものだけ全項目抽出し、10件表示せよ。
P-011	あいまい条件	後方一致	データの後方一致で条件指定する	★	顧客データ (df_customer) から顧客ID (customer_id) の末尾が1のものだけ全項目抽出し、10件表示せよ。
P-012	あいまい条件	部分一致	データの部分一致で条件指定する	★	店舗データ (df_store) から、住所 (address) に"横浜市"が含まれるものだけ全項目表示せよ。
P-013	あいまい条件	前方一致	正規表現の前方一致で条件指定する	★★	顧客データ (df_customer) から、ステータスコード (status_cd) の先頭がアルファベットのA～Fで始まるデータを全項目抽出し、10件表示せよ。
P-014	あいまい条件	後方一致	正規表現の後方一致で条件指定する	★★	顧客データ (df_customer) から、ステータスコード (status_cd) の末尾が数字の1～9で終わるデータを全項目抽出し、10件表示せよ。
P-015	あいまい条件	部分一致	正規表現の部分一致で条件指定する	★★	顧客データ (df_customer) から、ステータスコード (status_cd) の先頭がアルファベットのA～Fで始まり、末尾が数字の1～9で終わるデータを全項目抽出し、10件表示せよ。
P-016	あいまい条件	フォーマット一致	特定のデータ書式で条件指定する	★★	店舗データ (df_store) から、電話番号 (tel_no) が3桁-3桁-4桁のデータを全項目表示せよ。
P-017	ソート	並び替え	データを昇順に並べる	★	顧客データ (df_customer) を生年月日 (birth_day) で高齢順にソートし、先頭から全項目を10件表示せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
P-018	ソート	並び替え	データを降順に並べる	★	顧客データ（df_customer）を生年月日（birth_day）で若い順にソートし、先頭から全項目を10件表示せよ。
P-019	ソート	順位	順位付けする（同一順位あり）	★★	レシート明細データ（df_receipt）に対し、1件あたりの売上金額（amount）が高い順にランクを付与し、先頭から10件表示せよ。項目は顧客ID（customer_id）、売上金額（amount）、付与したランクを表示させること。なお、売上金額（amount）が等しい場合は同一順位を付与するものとする。
P-020	ソート	順位	順位付けする（同一順位なし）	★★	レシート明細データ（df_receipt）に対し、1件あたりの売上金額（amount）が高い順にランクを付与し、先頭から10件表示せよ。項目は顧客ID（customer_id）、売上金額（amount）、付与したランクを表示させること。なお、売上金額（amount）が等しい場合でも別順位を付与すること。
P-021	集計	カウント	データの件数をカウントする	★	レシート明細データ（df_receipt）に対し、件数をカウントせよ。
P-022	集計	カウント	データのユニーク件数をカウントする	★	レシート明細データ（df_receipt）の顧客ID（customer_id）に対し、ユニーク件数をカウントせよ。
P-023	集計	合計	対象データの合計値を算出する	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）と売上数量（quantity）を合計せよ。
P-024	集計	Max/Min	対象データの最大値を求める	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上年月日（sales_ymd）を求め、10件表示せよ。
P-025	集計	Max/Min	対象データの最小値を求める	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに最も古い売上年月日（sales_ymd）を求め、10件表示せよ。
P-026	集計	Max/Min	集計結果に対する条件指定で絞り込む	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上年月日（sales_ymd）と古い売上年月日を求め、両者が異なるデータを10件表示せよ。
P-027	集計	統計量	対象データの平均値を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の平均を計算し、降順でTOP5を表示せよ。
P-028	集計	統計量	対象データの中央値を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の中央値を計算し、降順でTOP5を表示せよ。
P-029	集計	統計量	対象データの最頻値を求める	★★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに商品コード（product_cd）の最頻値を求め、10件表示せよ。
P-030	集計	統計量	対象データの分散を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の分散を計算し、降順で5件表示せよ。
P-031	集計	統計量	対象データの標準偏差を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標準偏差を計算し、降順で5件表示せよ。
P-032	集計	統計量	データのパーセンタイル値を求める	★	レシート明細データ（df_receipt）の売上金額（amount）について、25%刻みでパーセンタイル値を求めよ。
P-033	集計	統計量	統計量に対する条件指定で絞り込む	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の平均を計算し、330以上のものを抽出せよ。
P-034	副問合せ	検索結果からのサブクエリ	検索結果から集計する	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに売上金額（amount）を合計して全顧客の平均を求めよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-035	副問合せ	条件指定でのサブクエリ	検索結果を条件指定に使った副問合せを行う	★★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに売上金額（amount）を合計して全顧客の平均を求め、平均以上に買い物をしている顧客を抽出し、10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-036	結合	内部結合	トランザクションとマスタを内部結合する	★	レシート明細データ（df_receipt）と店舗データ（df_store）を内部結合し、レシート明細データの全項目と店舗データの店舗名（store_name）を10件表示せよ。
P-037	結合	内部結合	マスタ同士を内部結合する	★	商品データ（df_product）とカテゴリデータ（df_category）を内部結合し、商品データの全項目とカテゴリデータのカテゴリ小区分名（category_small_name）を10件表示せよ。
P-038	結合	左外部結合	左外部結合でデータを残す	★	顧客データ（df_customer）とレシート明細データ（df_receipt）から、顧客ごとの売上金額合計を求め、10件表示せよ。ただし、売上実績がない顧客については売上金額を0として表示させること。また、顧客は性別コード（gender_cd）が女性（1）であるものを対象とし、非会員（顧客IDが"Z"から始まるもの）は除外すること。

No	大分類	小分類	設問主旨	難易度	設問内容
P-039	結合	完全外部結合	完全外部結合ですべてのレコードを残す	★	レシート明細データ (df_receipt) から、売上日数の多い顧客の上位20件を抽出したデータと、売上金額合計の多い顧客の上位20件を抽出したデータをそれぞれ作成し、さらにその2つを完全外部結合せよ。ただし、非会員（顧客IDが"Z"から始まるもの）は除外すること。
P-040	結合	クロス結合	クロス結合ですべてのレコードの組合せを作成する	★★	全ての店舗と全ての商品を組み合わせたデータを作成したい。店舗データ (df_store) と商品データ (df_product) を直横し、件数を計算せよ。
P-041	結合	自己結合による時系列のずらし	n件前のデータを結合する	★★	レシート明細データ (df_receipt) の売上金額 (amount) を日付 (sales_ymd) ごとに集計し、前回売上があった日からの売上金額増減を計算せよ。そして結果を10件表示せよ。
P-042	結合	自己結合による時系列のずらし	過去n件のデータを結合する	★★	レシート明細データ (df_receipt) の売上金額 (amount) を日付 (sales_ymd) ごとに集計し、各日付のデータに対し、前回、前々回、3回前に売上があった日のデータを結合せよ。そして結果を10件表示せよ。
P-043	縦横変換	縦から横への変換	縦持ちデータを横持ちデータに変換する	★	レシート明細データ (df_receipt) と顧客データ (df_customer) を結合し、性別コード (gender_cd) と年代 (ageから計算) ごとに売上金額 (amount) を合計した売上サマリデータを作成せよ。性別コードは0が男性、1が女性、9が不明を表すものとする。  ただし、項目構成は年代、女性の売上金額、男性の売上金額、性別不明の売上金額の4項目とすること（縦に年代、横に性別のクロス集計）。また、年代は10歳ごとの階級とすること。
P-044	縦横変換	横から縦への変換	横持ちデータを縦持ちデータに変換する	★	043で作成した売上サマリデータ (df_sales_summary) は性別の売上を横持ちさせたものであった。このデータから性別を縦持ちさせ、年代、性別コード、売上金額の3項目に変換せよ。ただし、性別コードは男性を"00"、女性を"01"、不明を"99"とする。
P-045	データ変換	日付型からの変換	日付型データを文字列データに変換する	★	顧客データ (df_customer) の生年月日 (birth_day) は日付型でデータを保有している。これをYYYYMMDD形式の文字列に変換し、顧客ID (customer_id) とともに10件表示せよ。
P-046	データ変換	日付型への変換	文字データを日付型データに変換する	★	顧客データ (df_customer) の申し込み日 (application_date) はYYYYMMDD形式の文字列型でデータを保有している。これを日付型に変換し、顧客ID (customer_id) とともに10件表示せよ。
P-047	データ変換	日付型への変換	数値データを日付型データに変換する	★	レシート明細データ (df_receipt) の売上エポック秒 (sales_epoch) は数値型のUNIX秒でデータを保有している。これを日付型に変換し、レシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。
P-048	データ変換	日付型への変換	エポック秒 (UNIX時間) を日付型データに変換する	★	レシート明細データフレーム (df_receipt) の売上エポック秒 (sales_epoch) は数値型のUNIX秒でデータを保有している。これを日付型に変換し、レシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。
P-049	データ変換	日付要素の取り出し	日付データから年だけ取り出す	★	レシート明細データ (df_receipt) の売上エポック秒 (sales_epoch) を日付型に変換し、「年」だけ取り出してレシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。
P-050	データ変換	日付要素の取り出し	日付データから月だけ取り出す	★	レシート明細データ (df_receipt) の売上エポック秒 (sales_epoch) を日付型に変換し、「月」だけ取り出してレシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。なお、「月」は0埋め2桁で取り出すこと。
P-051	データ変換	日付要素の取り出し	日付データから日だけ取り出す	★	レシート明細データ (df_receipt) の売上エポック秒を日付型に変換し、「日」だけ取り出してレシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。なお、「日」は0埋め2桁で取り出すこと。
P-052	データ変換	二値化	数値データを二値(0/1)データに変換する	★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客ID (customer_id) ごとに合計の上、売上金額合計に対して2,000円以下を0、2,000円より大きい金額を1に二値化し、顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-053	データ変換	二値化	文字データを二値(0/1)データに変換する	★★	顧客データ (df_customer) の郵便番号 (postal_cd) に対し、東京（先頭3桁が100～209のもの）を1、それ以外のものを0に二値化せよ。さらにレシート明細データ (df_receipt) と結合し、全期間において売上実績のある顧客数を、作成した二値ごとにカウントせよ。
P-054	データ変換	カテゴリ化	テキストラベルからカテゴリデータを作成する	★★	顧客データ (df_customer) の住所 (address) は、埼玉県、千葉県、東京都、神奈川県のおいずれかとなっている。都道府県毎にコード値を作成し、顧客ID、住所とともに10件表示せよ。値は埼玉県を11、千葉県を12、東京都を13、神奈川県を14とすること。

No	大分類	小分類	設問主旨	難易度	設問内容
P-055	データ変換	カテゴリ化	数値からカテゴリデータを作成する	★★	レシート明細（df_receipt）データの売上金額（amount）を顧客ID（customer_id）ごとに合計し、その合計金額の四分位点を求めよ。その上で、顧客ごとの売上金額合計に対して以下の基準でカテゴリ値を作成し、顧客ID、売上金額合計とともに10件表示せよ。カテゴリ値は順に1～4とする。  - 最小値以上第1四分位未満・・・1を付与 - 第1四分位以上第2四分位未満・・・2を付与 - 第2四分位以上第3四分位未満・・・3を付与 - 第3四分位以上・・・4を付与
P-056	データ変換	カテゴリ化	件数の少ないカテゴリを適切なカテゴリに寄せる	★★	顧客データ（df_customer）の年齢（age）をもとに10歳刻みで年代を算出し、顧客ID（customer_id）、生年月日（birth_day）とともに10件表示せよ。ただし、60歳以上は全て60歳代とすること。年代を表すカテゴリ名は任意とする。
P-057	データ変換	カテゴリ化	カテゴリ同士を組合せた新たなカテゴリを作成する	★	056の抽出結果と性別コード（gender_cd）により、新たに性別×年代の組み合わせを表すカテゴリデータを作成し、10件表示せよ。組み合わせを表すカテゴリの値は任意とする。
P-058	データ変換	ダミー変数化	ダミー変数(0/1)に変換する（カテゴリ型→ダミー変数）	★	顧客データ（df_customer）の性別コード（gender_cd）をダミー変数化し、顧客ID（customer_id）とともに10件表示せよ。
P-059	数値変換	正規化（z-score）	平均0、分散1に変換する	★	レシート明細データ（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を平均0、標準偏差1に標準化して顧客ID、売上金額合計とともに10件表示せよ。標準化に使用する標準偏差は、分散の平方根、もしくは不偏分散の平方根のどちらでも良いものとする。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-060	数値変換	正規化（Min-Max normalization）	最小値0、最大値1に変換する	★	レシート明細データ（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を最小値0、最大値1に正規化して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-061	数値変換	対数化	数値データを対数変換する（常用対数）	★	レシート明細データ（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を常用対数化（底10）して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-062	数値変換	対数化	数値データを対数変換する（自然対数）	★	レシート明細データ（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を自然対数化（底e）して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
P-063	四則演算	四則演算	数値を引き算する	★	商品データ（df_product）の単価（unit_price）と原価（unit_cost）から各商品の利益額を算出し、結果を10件表示せよ。
P-064	四則演算	四則演算	数値を割り算する	★	商品データ（df_product）の単価（unit_price）と原価（unit_cost）から、各商品の利益率の全体平均を算出せよ。ただし、単価と原価には欠損が生じていることに注意せよ。
P-065	四則演算	四則演算	除算結果に対して有効桁数以下を切り捨てる	★	商品データ（df_product）の各商品について、利益率が30%となる新たな単価を求めよ。ただし、1円未満は切り捨てること。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価（unit_price）と原価（unit_cost）には欠損が生じていることに注意せよ。
P-066	四則演算	小数の扱い	除算結果に対して有効桁数以下を四捨五入する	★	商品データ（df_product）の各商品について、利益率が30%となる新たな単価を求めよ。今回は、1円未満を丸めること（四捨五入または偶数への丸めで良い）。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価（unit_price）と原価（unit_cost）には欠損が生じていることに注意せよ。
P-067	四則演算	小数の扱い	除算結果に対して有効桁数以下を切り上げる	★	商品データ（df_product）の各商品について、利益率が30%となる新たな単価を求めよ。今回は、1円未満を切り上げること。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価（unit_price）と原価（unit_cost）には欠損が生じていることに注意せよ。
P-068	四則演算	小数の扱い	乗算結果に対して有効桁数以下を切り捨てる	★	商品データ（df_product）の各商品について、消費税率10%の税込み金額を求めよ。1円未満の端数は切り捨てとし、結果を10件表示せよ。ただし、単価（unit_price）には欠損が生じていることに注意せよ。
P-069	四則演算	集計結果の演算	集計結果から割合を計算する	★★	レシート明細データ（df_receipt）と商品データ（df_product）を結合し、顧客毎に全商品の売上金額合計と、カテゴリ大区分コード（category_major_cd）が"07"（瓶詰缶詰）の売上金額合計を計算の上、両者の比率を求めよ。抽出対象はカテゴリ大区分コード"07"（瓶詰缶詰）の売上実績がある顧客のみとし、結果を10件表示せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
P-070	日付型の計算	経過日数の計算	2つの日付から経過日数を計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、顧客データ (df_customer) の会員申込日 (application_date) からの経過日数を計算し、顧客ID (customer_id)、売上日、会員申込日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。
P-071	日付型の計算	経過日数の計算	2つの日付から経過月数を計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、顧客データ (df_customer) の会員申込日 (application_date) からの経過月数を計算し、顧客ID (customer_id)、売上日、会員申込日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。1ヶ月未満は切り捨てること。
P-072	日付型の計算	経過日数の計算	2つの日付から経過年数を計算する	★★	レシート明細データ (df_receipt) の売上日 (df_customer) に対し、顧客データ (df_customer) の会員申込日 (application_date) からの経過年数を計算し、顧客ID (customer_id)、売上日、会員申込日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。1年未満は切り捨てること。
P-073	日付型の計算	経過時間の計算	・2つの日付から経過時間をエポック秒で計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、顧客データ (df_customer) の会員申込日 (application_date) からのエポック秒による経過時間を計算し、顧客ID (customer_id)、売上日、会員申込日とともに10件表示せよ (なお、sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。なお、時間情報は保有していないため各日付は0時0分0秒を表すものとする。
P-074	日付型の計算	経過時間の計算	月曜日から経過日数を計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、当該週の月曜日からの経過日数を計算し、売上日、直前の月曜日付とともに10件表示せよ (sales_ymdは数値でデータを保持している点に注意)。
P-075	サンプリング	ランダム	ランダムサンプリングを行う(単純無作為抽出)	★	顧客データ (df_customer) からランダムに1%のデータを抽出し、先頭から10件表示せよ。
P-076	サンプリング	層化	カテゴリの割合に応じたサンプリングを行う(層化抽出)	★	顧客データ (df_customer) から性別コード (gender_cd) の割合に基づきランダムに10%のデータを層化抽出し、性別コードごとに件数を集計せよ。
P-077	外れ値・異常値	外れ値除外	統計的に外れ値を除外する (3σ外の除外)	★	レシート明細データ (df_receipt) の売上金額を顧客単位に合計し、合計した売上金額の外れ値を抽出せよ。なお、外れ値は売上金額合計を対数化したうえで平均と標準偏差を計算し、その平均から3σを超えて離れたものとする (自然対数と常用対数のどちらでも可)。結果は10件表示せよ。
P-078	外れ値・異常値	外れ値除外	統計的に外れ値を除外する (IQR1.5倍)	★★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客単位に合計し、合計した売上金額の外れ値を抽出せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。なお、ここでは外れ値を第1四分位と第3四分位の差であるIQRを用いて、「第1四分位数-1.5×IQR」を下回るもの、または「第3四分位数+1.5×IQR」を超えるものとする。結果は10件表示せよ。
P-079	欠損値	欠損列状況確認	欠損値がある列を確認する	★	商品データ (df_product) の各項目に対し、欠損数を確認せよ。
P-080	欠損値	欠損レコード削除	欠損値があるレコードを削除する	★	商品データ (df_product) のいずれかの項目に欠損が発生しているレコードを全て削除した新たな商品データを作成せよ。なお、削除前後の件数を表示させ、079で確認した件数だけ減少していることも確認すること。
P-081	欠損値	数値補完 (平均値)	平均値を用いて欠損値を補完する	★	単価 (unit_price) と原価 (unit_cost) の欠損値について、それぞれの平均値で補完した新たな商品データを作成せよ。なお、平均値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
P-082	欠損値	数値補完 (中央値)	中央値を用いて欠損値を補完する	★	単価 (unit_price) と原価 (unit_cost) の欠損値について、それぞれの中央値で補完した新たな商品データを作成せよ。なお、中央値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
P-083	欠損値	数値補完 (カテゴリごとの中央値)	カテゴリごとに算出した中央値で欠損値を補完する	★★	単価 (unit_price) と原価 (unit_cost) の欠損値について、各商品のカテゴリ小区分コード (category_small_cd) ごとに算出した中央値で補完した新たな商品データを作成せよ。なお、中央値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
P-084	除算エラー対応	0で代替	分母が欠損や0の場合でも除算結果データを作成する	★★	顧客データ (df_customer) の全顧客に対して全期間の売上金額に占める2019年売上金額の割合を計算し、新たなデータを作成せよ。ただし、売上実績がない場合は0として扱うこと。そして計算した割合が0超のものを抽出し、結果を10件表示せよ。また、作成したデータに欠損が存在しないことを確認せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
P-085	座標データ	ジオコード	郵便番号からジオコードに変換する	★	顧客データ (df_customer) の全顧客に対し、郵便番号 (postal_cd) を用いてジオコードデータ (df_geocode) を紐付け、新たな顧客データを作成せよ。ただし、1つの郵便番号 (postal_cd) に複数の経度 (longitude)、緯度 (latitude) 情報が紐づく場合は、経度 (longitude)、緯度 (latitude) の平均値を算出して使用すること。また、作成結果を確認するために結果を10件表示せよ。
P-086	座標データ	ジオコード	経度緯度から距離を計算する	★★★★	085で作成した緯度経度つき顧客データに対し、会員申込店舗コード (application_store_cd) をキーに店舗データ (df_store) と結合せよ。そして申込み店舗の緯度 (latitude) ・経度情報 (longitude) と顧客住所 (address) の緯度・経度を用いて申込み店舗と顧客住所の距離 (単位: km) を求め、顧客ID (customer_id)、顧客住所 (address)、店舗住所 (address) とともに表示せよ。計算式は以下の簡易式で良いものとするが、その他精度の高い方式を利用したライブラリを利用してもかまわない。結果は10件表示せよ。  緯度 (ラジアン) : $\phi$ 経度 (ラジアン) : $\lambda$ 距離 $L = 6371 * \arccos(\sin\phi_1 * \sin\phi_2 + \cos\phi_1 * \cos\phi_2 * \cos(\lambda_1 - \lambda_2))$
P-087	名寄せ	完全一致	PK以外の項目を利用した名寄せを行う	★★	顧客データ (df_customer) では、異なる店舗での申込みなどにより同一顧客が複数登録されている。名前 (customer_name) と郵便番号 (postal_cd) が同じ顧客は同一顧客とみなして1顧客1レコードとなるように名寄せした名寄顧客データを作成し、顧客データの件数、名寄顧客データの件数、重複数を算出せよ。ただし、同一顧客に対しては売上金額合計が最も高いものを残し、売上金額合計が同一もしくは売上実績がない顧客については顧客ID (customer_id) の番号が小さいものを残すこととする。
P-088	名寄せ	変換データ作成	名寄変換データを作成する	★★	087で作成したデータを元に、顧客データに統合名寄IDを付与したデータを作成せよ。ただし、統合名寄IDは以下の仕様で付与するものとする。  重複していない顧客: 顧客ID (customer_id) を設定 重複している顧客: 前設問で抽出したレコードの顧客IDを設定 顧客IDのユニーク件数と、統合名寄IDのユニーク件数の差も確認すること。
P-089	データ分割	レコードデータ	ホールドアウト法によるデータの分割を行う	★	売上実績がある顧客を、予測モデル構築のため学習用データとテスト用データに分割したい。それぞれ8:2の割合でランダムにデータを分割せよ。
P-090	データ分割	時系列データ	時系列データを分割する	★★	レシート明細データ (df_receipt) は2017年1月1日～2019年10月31日までのデータを有している。売上金額 (amount) を月次で集計し、学習用に12ヶ月、テスト用に6ヶ月の時系列モデル構築用データを3セット作成せよ。
P-091	不均衡データ	アンダーサンプリング	アンダーサンプリングにより不均衡データを調整する	★	顧客データ (df_customer) の各顧客に対し、売上実績がある顧客数と売上実績がない顧客数が1:1となるようにアンダーサンプリングで抽出せよ。
P-092	正規化・非正規化	正規化	第三正規形に正規化する	★	顧客データ (df_customer) の性別について、第三正規形へと正規化せよ。
P-093	正規化・非正規化	非正規化	第三正規形から非正規化する	★	商品データ (df_product) では各カテゴリのコード値だけを保有し、カテゴリ名は保有していない。カテゴリデータ (df_category) と組み合わせて非正規化し、カテゴリ名を保有した新たな商品データを作成せよ。
P-094	正規化・非正規化	CSV出力 (ヘッダ有り、コード変換なし)	ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する①	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。  - ファイル形式: CSV (カンマ区切り) - ヘッダ有無: 有り - 文字エンコーディング: UTF-8  ファイル出力先のパスは以下のようにすること。 - 出力先: ./data
P-095	ファイル入出力	CSV出力 (ヘッダ有り、UTF-8 -> SJIS)	ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する②	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。  - ファイル形式: CSV (カンマ区切り) - ヘッダ有無: 有り - 文字エンコーディング: CP932  ファイル出力先のパスは以下のようにすること。 - 出力先: ./data

No	大分類	小分類	設問主旨	難易度	設問内容
P-096	ファイル入出力	CSV出力（ヘッダ無し、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する③	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：無し - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること。 - 出力先：./data
P-097	ファイル入出力	CSV入力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを読み込む①	★	094で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8
P-098	ファイル入出力	CSV入力（ヘッダ無し、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを読み込む②	★	096で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：無し - 文字エンコーディング：UTF-8
P-099	ファイル入出力	TSV出力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、TSVファイルを作成する	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。 - ファイル形式：TSV（タブ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること。 - 出力先：./data
P-100	ファイル入出力	TSV入力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、TSVファイルを読み込む	★	099で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：TSV（タブ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8



No	大分類	小分類	設問主旨	難易度	設問内容
R-001	列に対する操作	全項目指定	全項目を指定行数抽出する	★	レシート明細データ（df_receipt）から全項目の先頭10件を表示し、どのようなデータを保有しているか目視で確認せよ。
R-002	列に対する操作	列指定	特定の列を抽出する	★	レシート明細データ（df_receipt）から売上年月日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上金額（amount）の順に列を指定し、10件表示せよ。
R-003	列に対する操作	列名変更	指定列の列名を変更する	★	レシート明細データ（df_receipt）から売上年月日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上金額（amount）の順に列を指定し、10件表示せよ。ただし、sales_ymdsales_dateに項目名を変更しながら抽出すること。
R-004	行に対する操作	単一条件	特定条件に合致する行を抽出(=,>,<)	★	レシート明細データ（df_receipt）から売上日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上金額（amount）の順に列を指定し、以下の条件を満たすデータを抽出せよ。  - 顧客ID（customer_id）が"CS018205000001"
R-005	行に対する操作	複数条件	複数条件に合致する行を抽出する①	★	レシート明細データ（df_receipt）から売上日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上金額（amount）の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID（customer_id）が"CS018205000001" - 売上金額（amount）が1,000以上
R-006	行に対する操作	複数条件	複数条件に合致する行を抽出する②	★	レシート明細データ（df_receipt）から売上日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上数量（quantity）、売上金額（amount）の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID（customer_id）が"CS018205000001" - 売上金額（amount）が1,000以上または売上数量（quantity）が5以上
R-007	行に対する操作	範囲指定	複数条件に合致する行を抽出する③	★	レシート明細データ（df_receipt）から売上日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上金額（amount）の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID（customer_id）が"CS018205000001" - 売上金額（amount）が1,000以上2,000以下
R-008	行に対する操作	不一致	特定条件に合致しない行を抽出する（!=）	★	レシート明細データ（df_receipt）から売上日（sales_ymd）、顧客ID（customer_id）、商品コード（product_cd）、売上金額（amount）の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID（customer_id）が"CS018205000001" - 商品コード（product_cd）が"P071401019"以外
R-009	行に対する操作	補集合	ド・モルガンの法則で書き換える	★	以下の処理において、出力結果を変えずにORをANDに書き換えよ。  df_store %>% filter(!(prefecture_cd == "13"   floor_area > 900))
R-010	あいまい条件	前方一致	データの前方一致で条件指定する	★	店舗データ（df_store）から、店舗コード（store_cd）が"S14"で始まるものだけ全項目抽出し、10件表示せよ。
R-011	あいまい条件	後方一致	データの後方一致で条件指定する	★	顧客データ（df_customer）から顧客ID（customer_id）の末尾が1のものだけ全項目抽出し、10件表示せよ。
R-012	あいまい条件	部分一致	データの部分一致で条件指定する	★	店舗データ（df_store）から、住所（address）に"横浜市"が含まれるものだけ全項目表示せよ。
R-013	あいまい条件	前方一致	正規表現の前方一致で条件指定する	★★	顧客データ（df_customer）から、ステータスコード（status_cd）の先頭がアルファベットのA～Fで始まるデータを全項目抽出し、10件表示せよ。
R-014	あいまい条件	後方一致	正規表現の後方一致で条件指定する	★★	顧客データ（df_customer）から、ステータスコード（status_cd）の末尾が数字の1～9で終わるデータを全項目抽出し、10件表示せよ。
R-015	あいまい条件	部分一致	正規表現の部分一致で条件指定する	★★	顧客データ（df_customer）から、ステータスコード（status_cd）の先頭がアルファベットのA～Fで始まり、末尾が数字の1～9で終わるデータを全項目抽出し、10件表示せよ。
R-016	あいまい条件	フォーマット一致	特定のデータ書式で条件指定する	★★	店舗データ（df_store）から、電話番号（tel_no）が3桁-3桁-4桁のデータを全項目表示せよ。
R-017	ソート	並び替え	データを昇順に並べる	★	顧客データ（df_customer）を生年月日（birth_day）で高齢順にソートし、先頭から全項目を10件表示せよ。



No	大分類	小分類	設問主旨	難易度	設問内容
R-018	ソート	並び替え	データを降順に並べる	★	顧客データ（df_customer）を生年月日（birth_day）で若い順にソートし、先頭から全項目を10件表示せよ。
R-019	ソート	順位	順位付けする（同一順位あり）	★★	レシート明細データ（df_receipt）に対し、1件あたりの売上金額（amount）が高い順にランクを付与し、先頭から10件表示せよ。項目は顧客ID（customer_id）、売上金額（amount）、付与したランクを表示させること。なお、売上金額（amount）が等しい場合は同一順位を付与するものとする。
R-020	ソート	順位	順位付けする（同一順位なし）	★★	レシート明細データ（df_receipt）に対し、1件あたりの売上金額（amount）が高い順にランクを付与し、先頭から10件表示せよ。項目は顧客ID（customer_id）、売上金額（amount）、付与したランクを表示させること。なお、売上金額（amount）が等しい場合でも別順位を付与すること。
R-021	集計	カウント	データの件数をカウントする	★	レシート明細データ（df_receipt）に対し、件数をカウントせよ。
R-022	集計	カウント	データのユニーク件数をカウントする	★	レシート明細データ（df_receipt）の顧客ID（customer_id）に対し、ユニーク件数をカウントせよ。
R-023	集計	合計	対象データの合計値を算出する	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）と売上数量（quantity）を合計せよ。
R-024	集計	Max/Min	対象データの最大値を求める	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上年月日（sales_ymd）を求め、10件表示せよ。
R-025	集計	Max/Min	対象データの最小値を求める	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに最も古い売上年月日（sales_ymd）を求め、10件表示せよ。
R-026	集計	Max/Min	集計結果に対する条件指定で絞り込む	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上年月日（sales_ymd）と古い売上年月日を求め、両者が異なるデータを10件表示せよ。
R-027	集計	統計量	対象データの平均値を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の平均を計算し、降順でTOP5を表示せよ。
R-028	集計	統計量	対象データの中央値を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の中央値を計算し、降順でTOP5を表示せよ。
R-029	集計	統計量	対象データの最頻値を求める	★★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに商品コード（product_cd）の最頻値を求め、10件表示させよ。
R-030	集計	統計量	対象データの分散を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の分散を計算し、降順で5件表示せよ。
R-031	集計	統計量	対象データの標準偏差を求める	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標準偏差を計算し、降順で5件表示せよ。
R-032	集計	統計量	データのパーセンタイル値を求める	★	レシート明細データ（df_receipt）の売上金額（amount）について、25%刻みでパーセンタイル値を求めよ。
R-033	集計	統計量	統計量に対する条件指定で絞り込む	★	レシート明細データ（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の平均を計算し、330以上のものを抽出せよ。
R-034	副問合せ	検索結果からのサブクエリ	検索結果から集計する	★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに売上金額（amount）を合計して全顧客の平均を求めよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-035	副問合せ	条件指定でのサブクエリ	検索結果を条件指定に使った副問合せを行う	★★	レシート明細データ（df_receipt）に対し、顧客ID（customer_id）ごとに売上金額（amount）を合計して全顧客の平均を求め、平均以上に買い物をしている顧客を抽出し、10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-036	結合	内部結合	トランザクションとマスタを内部結合する	★	レシート明細データ（df_receipt）と店舗データ（df_store）を内部結合し、レシート明細データの全項目と店舗データの店舗名（store_name）を10件表示せよ。
R-037	結合	内部結合	マスタ同士を内部結合する	★	商品データ（df_product）とカテゴリデータ（df_category）を内部結合し、商品データの全項目とカテゴリデータのカテゴリ小区分名（category_small_name）を10件表示せよ。
R-038	結合	左外部結合	左外部結合でデータを残す	★	顧客データ（df_customer）とレシート明細データ（df_receipt）から、顧客ごとの売上金額合計を求め、10件表示せよ。ただし、売上実績がない顧客については売上金額を0として表示させること。また、顧客は性別コード（gender_cd）が女性（1）であるものを対象とし、非会員（顧客IDが"Z"から始まるもの）は除外すること。

No	大分類	小分類	設問主旨	難易度	設問内容
R-039	結合	完全外部結合	完全外部結合ですべてのレコードを残す	★	レシート明細データ (df_receipt) から、売上日数の多い顧客の上位20件を抽出したデータと、売上金額合計の多い顧客の上位20件を抽出したデータをそれぞれ作成し、さらにその2つを完全外部結合せよ。ただし、非会員（顧客IDが"Z"から始まるもの）は除外すること。
R-040	結合	クロス結合	クロス結合ですべてのレコードの組合せを作成する	★★	全ての店舗と全ての商品を組み合わせたデータを作成したい。店舗データ (df_store) と商品データ (df_product) を直横し、件数を計算せよ。
R-041	結合	自己結合による時系列のずらし	n件前のデータを結合する	★★	レシート明細データ (df_receipt) の売上金額 (amount) を日付 (sales_ymd) ごとに集計し、前回売上があった日からの売上金額増減を計算せよ。そして結果を10件表示せよ。
R-042	結合	自己結合による時系列のずらし	過去n件のデータを結合する	★★	レシート明細データ (df_receipt) の売上金額 (amount) を日付 (sales_ymd) ごとに集計し、各日付のデータに対し、前回、前々回、3回前に売上があった日のデータを結合せよ。そして結果を10件表示せよ。
R-043	縦横変換	縦から横への変換	縦持ちデータを横持ちデータに変換する	★	レシート明細データ (df_receipt) と顧客データ (df_customer) を結合し、性別コード (gender_cd) と年代 (ageから計算) ごとに売上金額 (amount) を合計した売上サマリデータを作成せよ。性別コードは0が男性、1が女性、9が不明を表すものとする。  ただし、項目構成は年代、女性の売上金額、男性の売上金額、性別不明の売上金額の4項目とすること（縦に年代、横に性別のクロス集計）。また、年代は10歳ごとの階級とすること。
R-044	縦横変換	横から縦への変換	横持ちデータを縦持ちデータに変換する	★	043で作成した売上サマリデータ (df_sales_summary) は性別の売上を横持ちさせたものであった。このデータから性別を縦持ちさせ、年代、性別コード、売上金額の3項目に変換せよ。ただし、性別コードは男性を"00"、女性を"01"、不明を"99"とする。
R-045	データ変換	日付型からの変換	日付型データを文字列データに変換する	★	顧客データ (df_customer) の生年月日 (birth_day) は日付型でデータを保有している。これをYYYYMMDD形式の文字列に変換し、顧客ID (customer_id) とともに10件表示せよ。
R-046	データ変換	日付型への変換	文字データを日付型データに変換する	★	顧客データ (df_customer) の申し込み日 (application_date) はYYYYMMDD形式の文字列型でデータを保有している。これを日付型に変換し、顧客ID (customer_id) とともに10件表示せよ。
R-047	データ変換	日付型への変換	数値データを日付型データに変換する	★	レシート明細データ (df_receipt) の売上エポック秒 (sales_epoch) は数値型のUNIX秒でデータを保有している。これを日付型に変換し、レシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。
R-048	データ変換	日付型への変換	エポック秒 (UNIX時間) を日付型データに変換する	★	レシート明細データフレーム (df_receipt) の売上エポック秒 (sales_epoch) は数値型のUNIX秒でデータを保有している。これを日付型に変換し、レシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。
R-049	データ変換	日付要素の取り出し	日付データから年だけ取り出す	★	レシート明細データ (df_receipt) の売上エポック秒 (sales_epoch) を日付型に変換し、「年」だけ取り出してレシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。
R-050	データ変換	日付要素の取り出し	日付データから月だけ取り出す	★	レシート明細データ (df_receipt) の売上エポック秒 (sales_epoch) を日付型に変換し、「月」だけ取り出してレシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。なお、「月」は0埋め2桁で取り出すこと。
R-051	データ変換	日付要素の取り出し	日付データから日だけ取り出す	★	レシート明細データ (df_receipt) の売上エポック秒を日付型に変換し、「日」だけ取り出してレシート番号(receipt_no)、レシートサブ番号 (receipt_sub_no) とともに10件表示せよ。なお、「日」は0埋め2桁で取り出すこと。
R-052	データ変換	二値化	数値データを二値(0/1)データに変換する	★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客ID (customer_id) ごとに合計の上、売上金額合計に対して2,000円以下を0、2,000円より大きい金額を1に二値化し、顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-053	データ変換	二値化	文字データを二値(0/1)データに変換する	★★	顧客データ (df_customer) の郵便番号 (postal_cd) に対し、東京（先頭3桁が100～209のもの）を1、それ以外のものを0に二値化せよ。さらにレシート明細データ (df_receipt) と結合し、全期間において売上実績のある顧客数を、作成した二値ごとにカウントせよ。
R-054	データ変換	カテゴリ化	テキストラベルからカテゴリデータを作成する	★★	顧客データ (df_customer) の住所 (address) は、埼玉県、千葉県、東京都、神奈川県のおいずれかとなっている。都道府県毎にコード値を作成し、顧客ID、住所とともに10件表示せよ。値は埼玉県を11、千葉県を12、東京都を13、神奈川県を14とすること。

No	大分類	小分類	設問主旨	難易度	設問内容
R-055	データ変換	カテゴリ化	数値からカテゴリデータを作成する	★★	レシート明細 (df_receipt) データの売上金額 (amount) を顧客ID (customer_id) ごとに合計し、その合計金額の四分位点を求めよ。その上で、顧客ごとの売上金額合計に対して以下の基準でカテゴリ値を作成し、顧客ID、売上金額合計とともに10件表示せよ。カテゴリ値は順に1~4とする。  - 最小値以上第1四分位未満・・・1を付与 - 第1四分位以上第2四分位未満・・・2を付与 - 第2四分位以上第3四分位未満・・・3を付与 - 第3四分位以上・・・4を付与
R-056	データ変換	カテゴリ化	件数の少ないカテゴリを適切なカテゴリに寄せる	★★	顧客データ (df_customer) の年齢 (age) をもとに10歳刻みで年代を算出し、顧客ID (customer_id)、生年月日 (birth_day) とともに10件表示せよ。ただし、60歳以上は全て60歳代とすること。年代を表すカテゴリ名は任意とする。
R-057	データ変換	カテゴリ化	カテゴリ同士を組合せた新たなカテゴリを作成する	★	056の抽出結果と性別コード (gender_cd) により、新たに性別×年代の組み合わせを表すカテゴリデータを作成し、10件表示せよ。組み合わせを表すカテゴリの値は任意とする。
R-058	データ変換	ダミー変数化	ダミー変数(0/1)に変換する (カテゴリ型→ダミー変数)	★	顧客データ (df_customer) の性別コード (gender_cd) をダミー変数化し、顧客ID (customer_id) とともに10件表示せよ。
R-059	数値変換	正規化 (z-score)	平均0、分散1に変換する	★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客ID (customer_id) ごとに合計し、売上金額合計を平均0、標準偏差1に標準化して顧客ID、売上金額合計とともに10件表示せよ。標準化に使用する標準偏差は、分散の平方根、もしくは不偏分散の平方根のどちらでも良いものとする。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-060	数値変換	正規化 (Min-Max normalization)	最小値0、最大値1に変換する	★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客ID (customer_id) ごとに合計し、売上金額合計を最小値0、最大値1に正規化して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-061	数値変換	対数化	数値データを対数変換する (常用対数)	★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客ID (customer_id) ごとに合計し、売上金額合計を常用対数化 (底10) して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-062	数値変換	対数化	数値データを対数変換する (自然対数)	★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客ID (customer_id) ごとに合計し、売上金額合計を自然対数化 (底e) して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
R-063	四則演算	四則演算	数値を引き算する	★	商品データ (df_product) の単価 (unit_price) と原価 (unit_cost) から各商品の利益額を算出し、結果を10件表示せよ。
R-064	四則演算	四則演算	数値を割り算する	★	商品データ (df_product) の単価 (unit_price) と原価 (unit_cost) から、各商品の利益率の全体平均を算出せよ。ただし、単価と原価には欠損が生じていることに注意せよ。
R-065	四則演算	四則演算	除算結果に対して有効桁数以下を切り捨てる	★	商品データ (df_product) の各商品について、利益率が30%となる新たな単価を求めよ。ただし、1円未満は切り捨てること。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価 (unit_price) と原価 (unit_cost) には欠損が生じていることに注意せよ。
R-066	四則演算	小数の扱い	除算結果に対して有効桁数以下を四捨五入する	★	商品データ (df_product) の各商品について、利益率が30%となる新たな単価を求めよ。今回は、1円未満を丸めること (四捨五入または偶数への丸めで良い)。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価 (unit_price) と原価 (unit_cost) には欠損が生じていることに注意せよ。
R-067	四則演算	小数の扱い	除算結果に対して有効桁数以下を切り上げる	★	商品データ (df_product) の各商品について、利益率が30%となる新たな単価を求めよ。今回は、1円未満を切り上げること。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価 (unit_price) と原価 (unit_cost) には欠損が生じていることに注意せよ。
R-068	四則演算	小数の扱い	乗算結果に対して有効桁数以下を切り捨てる	★	商品データ (df_product) の各商品について、消費税率10%の税込み金額を求めよ。1円未満の端数は切り捨てとし、結果を10件表示せよ。ただし、単価 (unit_price) には欠損が生じていることに注意せよ。
R-069	四則演算	集計結果の演算	集計結果から割合を計算する	★★	レシート明細データ (df_receipt) と商品データ (df_product) を結合し、顧客毎に全商品の売上金額合計と、カテゴリ大区分コード (category_major_cd) が"07" (瓶詰缶詰) の売上金額合計を計算の上、両者の比率を求めよ。抽出対象はカテゴリ大区分コード"07" (瓶詰缶詰) の売上実績がある顧客のみとし、結果を10件表示せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
R-070	日付型の計算	経過日数の計算	2つの日付から経過日数を計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、顧客データ (df_customer) の会員登録日 (application_date) からの経過日数を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。
R-071	日付型の計算	経過日数の計算	2つの日付から経過月数を計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、顧客データ (df_customer) の会員登録日 (application_date) からの経過月数を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。1ヶ月未満は切り捨てること。
R-072	日付型の計算	経過日数の計算	2つの日付から経過年数を計算する	★★	レシート明細データ (df_receipt) の売上日 (df_customer) に対し、顧客データ (df_customer) の会員登録日 (application_date) からの経過年数を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。1年未満は切り捨てること。
R-073	日付型の計算	経過時間の計算	・2つの日付から経過時間をエポック秒で計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、顧客データ (df_customer) の会員登録日 (application_date) からのエポック秒による経過時間を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (なお、sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。なお、時間情報は保有していないため各日付は0時0分0秒を表すものとする。
R-074	日付型の計算	経過時間の計算	月曜日から経過日数を計算する	★★	レシート明細データ (df_receipt) の売上日 (sales_ymd) に対し、当該週の月曜日からの経過日数を計算し、売上日、直前の月曜日付とともに10件表示せよ (sales_ymdは数値でデータを保持している点に注意)。
R-075	サンプリング	ランダム	ランダムサンプリングを行う(単純無作為抽出)	★	顧客データ (df_customer) からランダムに1%のデータを抽出し、先頭から10件表示せよ。
R-076	サンプリング	層化	カテゴリの割合に応じたサンプリングを行う(層化抽出)	★	顧客データ (df_customer) から性別コード (gender_cd) の割合に基づきランダムに10%のデータを層化抽出し、性別コードごとに件数を集計せよ。
R-077	外れ値・異常値	外れ値除外	統計的に外れ値を除外する (3σ外の除外)	★	レシート明細データ (df_receipt) の売上金額を顧客単位に合計し、合計した売上金額の外れ値を抽出せよ。なお、外れ値は売上金額合計を対数化したうえで平均と標準偏差を計算し、その平均から3σを超えて離れたものとする (自然対数と常用対数のどちらでも可)。結果は10件表示せよ。
R-078	外れ値・異常値	外れ値除外	統計的に外れ値を除外する (IQR1.5倍)	★★	レシート明細データ (df_receipt) の売上金額 (amount) を顧客単位に合計し、合計した売上金額の外れ値を抽出せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。なお、ここでは外れ値を第1四分位と第3四分位の差であるIQRを用いて、「第1四分位数-1.5×IQR」を下回るもの、または「第3四分位数+1.5×IQR」を超えるものとする。結果は10件表示せよ。
R-079	欠損値	欠損列状況確認	欠損値がある列を確認する	★	商品データ (df_product) の各項目に対し、欠損数を確認せよ。
R-080	欠損値	欠損レコード削除	欠損値があるレコードを削除する	★	商品データ (df_product) のいずれかの項目に欠損が発生しているレコードを全て削除した新たな商品データを作成せよ。なお、削除前後の件数を表示させ、079で確認した件数だけ減少していることも確認すること。
R-081	欠損値	数値補完 (平均値)	平均値を用いて欠損値を補完する	★	単価 (unit_price) と原価 (unit_cost) の欠損値について、それぞれの平均値で補完した新たな商品データを作成せよ。なお、平均値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
R-082	欠損値	数値補完 (中央値)	中央値を用いて欠損値を補完する	★	単価 (unit_price) と原価 (unit_cost) の欠損値について、それぞれの中央値で補完した新たな商品データを作成せよ。なお、中央値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
R-083	欠損値	数値補完 (カテゴリごとの中央値)	カテゴリごとに算出した中央値で欠損値を補完する	★★	単価 (unit_price) と原価 (unit_cost) の欠損値について、各商品のカテゴリ小区分コード (category_small_cd) ごとに算出した中央値で補完した新たな商品データを作成せよ。なお、中央値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
R-084	除算エラー対応	0で代替	分母が欠損や0の場合でも除算結果データを作成する	★★	顧客データ (df_customer) の全顧客に対して全期間の売上金額に占める2019年売上金額の割合を計算し、新たなデータを作成せよ。ただし、売上実績がない場合は0として扱うこと。そして計算した割合が0超のものを抽出し、結果を10件表示せよ。また、作成したデータに欠損が存在しないことを確認せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
R-085	座標データ	ジオコード	郵便番号からジオコードに変換する	★	顧客データ（df_customer）の全顧客に対し、郵便番号（postal_cd）を用いてジオコードデータ（df_geocode）を紐付け、新たな顧客データを作成せよ。ただし、1つの郵便番号（postal_cd）に複数の経度（longitude）、緯度（latitude）情報が紐づく場合は、経度（longitude）、緯度（latitude）の平均値を算出して使用すること。また、作成結果を確認するために結果を10件表示せよ。
R-086	座標データ	ジオコード	経度緯度から距離を計算する	★★★★	085で作成した緯度経度つき顧客データに対し、会員申込店舗コード（application_store_cd）をキーに店舗データ（df_store）と結合せよ。そして申込み店舗の緯度（latitude）・経度情報（longitude）と顧客住所（address）の緯度・経度を用いて申込み店舗と顧客住所の距離（単位：km）を求め、顧客ID（customer_id）、顧客住所（address）、店舗住所（address）とともに表示せよ。計算式は以下の簡易式で良いものとするが、その他精度の高い方式を利用したライブラリを利用してもかまわない。結果は10件表示せよ。  緯度（ラジアン）： $\phi$ 経度（ラジアン）： $\lambda$ 距離 $L=6371 \cdot \arccos(\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\lambda_1 - \lambda_2))$
R-087	名寄せ	完全一致	PK以外の項目を利用した名寄せを行う	★★	顧客データ（df_customer）では、異なる店舗での申込みなどにより同一顧客が複数登録されている。名前（customer_name）と郵便番号（postal_cd）が同じ顧客は同一顧客とみなして1顧客1レコードとなるように名寄せした名寄顧客データを作成し、顧客データの件数、名寄顧客データの件数、重複数を算出せよ。ただし、同一顧客に対しては売上金額合計が最も高いものを残し、売上金額合計が同一もしくは売上実績がない顧客については顧客ID（customer_id）の番号が小さいものを残すこととする。
R-088	名寄せ	変換データ作成	名寄変換データを作成する	★★	087で作成したデータを元に、顧客データに統合名寄IDを付与したデータを作成せよ。ただし、統合名寄IDは以下の仕様で付与するものとする。  重複していない顧客：顧客ID（customer_id）を設定 重複している顧客：前設問で抽出したレコードの顧客IDを設定 顧客IDのユニーク件数と、統合名寄IDのユニーク件数の差も確認すること。
R-089	データ分割	レコードデータ	ホールドアウト法によるデータの分割を行う	★	売上実績がある顧客を、予測モデル構築のため学習用データとテスト用データに分割したい。それぞれ8:2の割合でランダムにデータを分割せよ。
R-090	データ分割	時系列データ	時系列データを分割する	★★	レシート明細データ（df_receipt）は2017年1月1日～2019年10月31日までのデータを有している。売上金額（amount）を月次で集計し、学習用に12ヶ月、テスト用に6ヶ月の時系列モデル構築用データを3セット作成せよ。
R-091	不均衡データ	アンダーサンプリング	アンダーサンプリングにより不均衡データを調整する	★	顧客データ（df_customer）の各顧客に対し、売上実績がある顧客数と売上実績がない顧客数が1:1となるようにアンダーサンプリングで抽出せよ。
R-092	正規化・非正規化	正規化	第三正規形に正規化する	★	顧客データ（df_customer）の性別について、第三正規形へと正規化せよ。
R-093	正規化・非正規化	非正規化	第三正規形から非正規化する	★	商品データ（df_product）では各カテゴリのコード値だけを保有し、カテゴリ名は保有していない。カテゴリデータ（df_category）と組み合わせて非正規化し、カテゴリ名を保有した新たな商品データを作成せよ。
R-094	正規化・非正規化	CSV出力（ヘッダ有り、コード変換なし）	ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する①	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること。 - 出力先：./data
R-095	ファイル入出力	CSV出力（ヘッダ有り、UTF-8 -> SJIS）	ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する②	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：CP932  ファイル出力先のパスは以下のようにすること。 - 出力先：./data

No	大分類	小分類	設問主旨	難易度	設問内容
R-096	ファイル入出力	CSV出力（ヘッダ無し、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する③	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：無し - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること。 - 出力先：./data
R-097	ファイル入出力	CSV入力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを読み込む①	★	094で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8
R-098	ファイル入出力	CSV入力（ヘッダ無し、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを読み込む②	★	096で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：無し - 文字エンコーディング：UTF-8
R-099	ファイル入出力	TSV出力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、TSVファイルを作成する	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。 - ファイル形式：TSV（タブ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること。 - 出力先：./data
R-100	ファイル入出力	TSV入力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、TSVファイルを読み込む	★	099で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：TSV（タブ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8

No	大分類	小分類	設問主旨	難易度	設問内容
S-001	列に対する操作	全項目指定	全項目を指定行数抽出する	★	レシート明細データ (receipt) から全項目の先頭10件を表示し、どのようなデータを保有しているか目視で確認せよ。
S-002	列に対する操作	列指定	特定の列を抽出する	★	レシート明細データ (receipt) から売上年月日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、10件表示せよ。
S-003	列に対する操作	列名変更	指定列の列名を変更する	★	レシート明細データ (receipt) から売上年月日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、10件表示せよ。ただし、sales_ymdsales_dateに項目名を変更しながら抽出すること。
S-004	行に対する操作	単一条件	特定条件に合致する行を抽出(=,>,<)	★	レシート明細データ (receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001"
S-005	行に対する操作	複数条件	複数条件に合致する行を抽出する①	★	レシート明細データ (receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 売上金額 (amount) が1,000以上
S-006	行に対する操作	複数条件	複数条件に合致する行を抽出する②	★	レシート明細データ (receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上数量 (quantity)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 売上金額 (amount) が1,000以上または売上数量 (quantity) が5以上
S-007	行に対する操作	範囲指定	複数条件に合致する行を抽出する③	★	レシート明細データ (receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 売上金額 (amount) が1,000以上2,000以下
S-008	行に対する操作	不一致	特定条件に合致しない行を抽出する (!=)	★	レシート明細データ (receipt) から売上日 (sales_ymd)、顧客ID (customer_id)、商品コード (product_cd)、売上金額 (amount) の順に列を指定し、以下の全ての条件を満たすデータを抽出せよ。  - 顧客ID (customer_id) が"CS018205000001" - 商品コード (product_cd) が"P071401019"以外
S-009	行に対する操作	補集合	ド・モルガンの法則で書き換える	★	以下の処理において、出力結果を変えずにORをANDに書き換えよ。  SELECT * FROM store WHERE NOT (prefecture_cd = '13' OR floor_area > 900)
S-010	あいまい条件	前方一致	データの前方一致で条件指定する	★	店舗データ (store) から、店舗コード (store_cd) が"S14"で始まるものだけ全項目抽出し、10件表示せよ。
S-011	あいまい条件	後方一致	データの後方一致で条件指定する	★	顧客データ (customer) から顧客ID (customer_id) の末尾が1のものだけ全項目抽出し、10件表示せよ。
S-012	あいまい条件	部分一致	データの部分一致で条件指定する	★	店舗データ (store) から、住所 (address) に"横浜市"が含まれるものだけ全項目表示せよ。
S-013	あいまい条件	前方一致	正規表現の前方一致で条件指定する	★★	顧客データ (customer) から、ステータスコード (status_cd) の先頭がアルファベットのA～Fで始まるデータを全項目抽出し、10件表示せよ。
S-014	あいまい条件	後方一致	正規表現の後方一致で条件指定する	★★	顧客データ (customer) から、ステータスコード (status_cd) の末尾が数字の1～9で終わるデータを全項目抽出し、10件表示せよ。
S-015	あいまい条件	部分一致	正規表現の部分一致で条件指定する	★★	顧客データ (customer) から、ステータスコード (status_cd) の先頭がアルファベットのA～Fで始まり、末尾が数字の1～9で終わるデータを全項目抽出し、10件表示せよ。
S-016	あいまい条件	フォーマット一致	特定のデータ書式で条件指定する	★★	店舗データ (store) から、電話番号 (tel_no) が3桁-3桁-4桁のデータを全項目表示せよ。
S-017	ソート	並び替え	データを昇順に並べる	★	顧客データ (customer) を生年月日 (birth_day) で高齢順にソートし、先頭から全項目を10件表示せよ。



No	大分類	小分類	設問主旨	難易度	設問内容
S-018	ソート	並び替え	データを降順に並べる	★	顧客データ（customer）を生年月日（birth_day）で若い順にソートし、先頭から全項目を10件表示せよ。
S-019	ソート	順位	順位付けする（同一順位あり）	★★	レシート明細データ（receipt）に対し、1件あたりの売上金額（amount）が高い順にランクを付与し、先頭から10件表示せよ。項目は顧客ID（customer_id）、売上金額（amount）、付与したランクを表示させること。なお、売上金額（amount）が等しい場合は同一順位を付与するものとする。
S-020	ソート	順位	順位付けする（同一順位なし）	★★	レシート明細データ（receipt）に対し、1件あたりの売上金額（amount）が高い順にランクを付与し、先頭から10件表示せよ。項目は顧客ID（customer_id）、売上金額（amount）、付与したランクを表示させること。なお、売上金額（amount）が等しい場合でも別順位を付与すること。
S-021	集計	カウント	データの件数をカウントする	★	レシート明細データ（receipt）に対し、件数をカウントせよ。
S-022	集計	カウント	データのユニーク件数をカウントする	★	レシート明細データ（receipt）の顧客ID（customer_id）に対し、ユニーク件数をカウントせよ。
S-023	集計	合計	対象データの合計値を算出する	★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）と売上数量（quantity）を合計せよ。
S-024	集計	Max/Min	対象データの最大値を求める	★	レシート明細データ（receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上年月日（sales_ymd）を求め、10件表示せよ。
S-025	集計	Max/Min	対象データの最小値を求める	★	レシート明細データ（receipt）に対し、顧客ID（customer_id）ごとに最も古い売上年月日（sales_ymd）を求め、10件表示せよ。
S-026	集計	Max/Min	集計結果に対する条件指定で絞り込む	★	レシート明細データ（receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上年月日（sales_ymd）と古い売上年月日を求め、両者が異なるデータを10件表示せよ。
S-027	集計	統計量	対象データの平均値を求める	★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の平均を計算し、降順でTOP5を表示せよ。
S-028	集計	統計量	対象データの中央値を求める	★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の中央値を計算し、降順でTOP5を表示せよ。
S-029	集計	統計量	対象データの最頻値を求める	★★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに商品コード（product_cd）の最頻値を求め、10件表示させよ。
S-030	集計	統計量	対象データの分散を求める	★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の分散を計算し、降順で5件表示せよ。
S-031	集計	統計量	対象データの標準偏差を求める	★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標準偏差を計算し、降順で5件表示せよ。
S-032	集計	統計量	データのパーセンタイル値を求める	★	レシート明細データ（receipt）の売上金額（amount）について、25%刻みでパーセンタイル値を求めよ。
S-033	集計	統計量	統計量に対する条件指定で絞り込む	★	レシート明細データ（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の平均を計算し、330以上のものを抽出せよ。
S-034	副問合せ	検索結果からのサブクエリ	検索結果から集計する	★	レシート明細データ（receipt）に対し、顧客ID（customer_id）ごとに売上金額（amount）を合計して全顧客の平均を求めよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-035	副問合せ	条件指定でのサブクエリ	検索結果を条件指定に使った副問合せを行う	★★	レシート明細データ（receipt）に対し、顧客ID（customer_id）ごとに売上金額（amount）を合計して全顧客の平均を求め、平均以上に買い物をしている顧客を抽出し、10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-036	結合	内部結合	トランザクションとマスタを内部結合する	★	レシート明細データ（receipt）と店舗データ（store）を内部結合し、レシート明細データの全項目と店舗データの店舗名（store_name）を10件表示せよ。
S-037	結合	内部結合	マスタ同士を内部結合する	★	商品データ（product）とカテゴリデータ（category）を内部結合し、商品データの全項目とカテゴリデータのカテゴリ小区分名（category_small_name）を10件表示せよ。
S-038	結合	左外部結合	左外部結合でデータを残す	★	顧客データ（customer）とレシート明細データ（receipt）から、顧客ごとの売上金額合計を求め、10件表示せよ。ただし、売上実績がない顧客については売上金額を0として表示させること。また、顧客は性別コード（gender_cd）が女性（1）であるものを対象とし、非会員（顧客IDが"Z"から始まるもの）は除外すること。

No	大分類	小分類	設問主旨	難易度	設問内容
S-039	結合	完全外部結合	完全外部結合ですべてのレコードを残す	★	レシート明細データ（receipt）から、売上日数の多い顧客の上位20件を抽出したデータと、売上金額合計の多い顧客の上位20件を抽出したデータをそれぞれ作成し、さらにその2つを完全外部結合せよ。ただし、非会員（顧客IDが"Z"から始まるもの）は除外すること。
S-040	結合	クロス結合	クロス結合ですべてのレコードの組合せを作成する	★★	全ての店舗と全ての商品を組み合わせたデータを作成したい。店舗データ（store）と商品データ（product）を直積し、件数を計算せよ。
S-041	結合	自己結合による時系列のずらし	n件前のデータを結合する	★★	レシート明細データ（receipt）の売上金額（amount）を日付（sales_ymd）ごとに集計し、前回売上があった日からの売上金額増減を計算せよ。そして結果を10件表示せよ。
S-042	結合	自己結合による時系列のずらし	過去n件のデータを結合する	★★	レシート明細データ（receipt）の売上金額（amount）を日付（sales_ymd）ごとに集計し、各日付のデータに対し、前回、前々回、3回前に売上があった日のデータを結合せよ。そして結果を10件表示せよ。
S-043	縦横変換	縦から横への変換	縦持ちデータを横持ちデータに変換する	★	レシート明細データ（receipt）と顧客データ（customer）を結合し、性別コード（gender_cd）と年代（ageから計算）ごとに売上金額（amount）を合計した売上サマリデータを作成せよ。性別コードは0が男性、1が女性、9が不明を表すものとする。  ただし、項目構成は年代、女性の売上金額、男性の売上金額、性別不明の売上金額の4項目とすること（縦に年代、横に性別のクロス集計）。また、年代は10歳ごとの階級とすること。
S-044	縦横変換	横から縦への変換	横持ちデータを縦持ちデータに変換する	★	043で作成した売上サマリデータ（sales_summary）は性別の売上を横持ちさせたものであった。このデータから性別を縦持ちさせ、年代、性別コード、売上金額の3項目に変換せよ。ただし、性別コードは男性を"00"、女性を"01"、不明を"99"とする。
S-045	データ変換	日付型からの変換	日付型データを文字列データに変換する	★	顧客データ（customer）の生年月日（birth_day）は日付型でデータを保有している。これをYYYYMMDD形式の文字列に変換し、顧客ID（customer_id）とともに10件表示せよ。
S-046	データ変換	日付型への変換	文字データを日付型データに変換する	★	顧客データ（customer）の申し込み日（application_date）はYYYYMMDD形式の文字列型でデータを保有している。これを日付型に変換し、顧客ID（customer_id）とともに10件表示せよ。
S-047	データ変換	日付型への変換	数値データを日付型データに変換する	★	レシート明細データ（receipt）の売上エポック秒（sales_epoch）は数値型のUNIX秒でデータを保有している。これを日付型に変換し、レシート番号（receipt_no）、レシートサブ番号（receipt_sub_no）とともに10件表示せよ。
S-048	データ変換	日付型への変換	エポック秒（UNIX時間）を日付型データに変換する	★	レシート明細データフレーム（receipt）の売上エポック秒（sales_epoch）は数値型のUNIX秒でデータを保有している。これを日付型に変換し、レシート番号（receipt_no）、レシートサブ番号（receipt_sub_no）とともに10件表示せよ。
S-049	データ変換	日付要素の取り出し	日付データから年だけ取り出す	★	レシート明細データ（receipt）の売上エポック秒（sales_epoch）を日付型に変換し、「年」だけ取り出してレシート番号（receipt_no）、レシートサブ番号（receipt_sub_no）とともに10件表示せよ。
S-050	データ変換	日付要素の取り出し	日付データから月だけ取り出す	★	レシート明細データ（receipt）の売上エポック秒（sales_epoch）を日付型に変換し、「月」だけ取り出してレシート番号（receipt_no）、レシートサブ番号（receipt_sub_no）とともに10件表示せよ。なお、「月」は0埋め2桁で取り出すこと。
S-051	データ変換	日付要素の取り出し	日付データから日だけ取り出す	★	レシート明細データ（receipt）の売上エポック秒を日付型に変換し、「日」だけ取り出してレシート番号（receipt_no）、レシートサブ番号（receipt_sub_no）とともに10件表示せよ。なお、「日」は0埋め2桁で取り出すこと。
S-052	データ変換	二値化	数値データを二値(0/1)データに変換する	★	レシート明細データ（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計の上、売上金額合計に対して2,000円以下を0、2,000円より大きい金額を1に二値化し、顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-053	データ変換	二値化	文字データを二値(0/1)データに変換する	★★	顧客データ（customer）の郵便番号（postal_cd）に対し、東京（先頭3桁が100～209のもの）を1、それ以外のものを0に二値化せよ。さらにレシート明細データ（receipt）と結合し、全期間において売上実績のある顧客数を、作成した二値ごとにカウントせよ。
S-054	データ変換	カテゴリ化	テキストラベルからカテゴリデータを作成する	★★	顧客データ（customer）の住所（address）は、埼玉県、千葉県、東京都、神奈川県のがずれかとなっている。都道府県毎にコード値を作成し、顧客ID、住所とともに10件表示せよ。値は埼玉県を11、千葉県を12、東京都を13、神奈川県を14とすること。

No	大分類	小分類	設問主旨	難易度	設問内容
S-055	データ変換	カテゴリ化	数値からカテゴリデータを作成する	★★	レシート明細（receipt）データの売上金額（amount）を顧客ID（customer_id）ごとに合計し、その合計金額の四分位点を求めよ。その上で、顧客ごとの売上金額合計に対して以下の基準でカテゴリ値を作成し、顧客ID、売上金額合計とともに10件表示せよ。カテゴリ値は順に1～4とする。  - 最小値以上第1四分位未満・・・1を付与 - 第1四分位以上第2四分位未満・・・2を付与 - 第2四分位以上第3四分位未満・・・3を付与 - 第3四分位以上・・・4を付与
S-056	データ変換	カテゴリ化	件数の少ないカテゴリを適切なカテゴリに寄せる	★★	顧客データ（customer）の年齢（age）をもとに10歳刻みで年代を算出し、顧客ID（customer_id）、生年月日（birth_day）とともに10件表示せよ。ただし、60歳以上は全て60歳代とすること。年代を表すカテゴリ名は任意とする。
S-057	データ変換	カテゴリ化	カテゴリ同士を組合せた新たなカテゴリを作成する	★	056の抽出結果と性別コード（gender_cd）により、新たに性別×年代の組み合わせを表すカテゴリデータを作成し、10件表示せよ。組み合わせを表すカテゴリの値は任意とする。
S-058	データ変換	ダミー変数化	ダミー変数(0/1)に変換する（カテゴリ型→ダミー変数）	★	顧客データ（customer）の性別コード（gender_cd）をダミー変数化し、顧客ID（customer_id）とともに10件表示せよ。
S-059	数値変換	正規化（z-score）	平均0、分散1に変換する	★	レシート明細データ（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を平均0、標準偏差1に標準化して顧客ID、売上金額合計とともに10件表示せよ。標準化に使用する標準偏差は、分散の平方根、もしくは不偏分散の平方根のどちらでも良いものとする。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-060	数値変換	正規化（Min-Max normalization）	最小値0、最大値1に変換する	★	レシート明細データ（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を最小値0、最大値1に正規化して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-061	数値変換	対数化	数値データを対数変換する（常用対数）	★	レシート明細データ（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を常用対数化（底10）して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-062	数値変換	対数化	数値データを対数変換する（自然対数）	★	レシート明細データ（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、売上金額合計を自然対数化（底e）して顧客ID、売上金額合計とともに10件表示せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。
S-063	四則演算	四則演算	数値を引き算する	★	商品データ（product）の単価（unit_price）と原価（unit_cost）から各商品の利益額を算出し、結果を10件表示せよ。
S-064	四則演算	四則演算	数値を割り算する	★	商品データ（product）の単価（unit_price）と原価（unit_cost）から、各商品の利益率の全体平均を算出せよ。ただし、単価と原価には欠損が生じていることに注意せよ。
S-065	四則演算	四則演算	除算結果に対して有効桁数以下を切り捨てる	★	商品データ（product）の各商品について、利益率が30%となる新たな単価を求めよ。ただし、1円未満は切り捨てること。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価（unit_price）と原価（unit_cost）には欠損が生じていることに注意せよ。
S-066	四則演算	小数の扱い	除算結果に対して有効桁数以下を四捨五入する	★	商品データ（product）の各商品について、利益率が30%となる新たな単価を求めよ。今回は、1円未満を丸めること（四捨五入または偶数への丸めで良い）。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価（unit_price）と原価（unit_cost）には欠損が生じていることに注意せよ。
S-067	四則演算	小数の扱い	除算結果に対して有効桁数以下を切り上げる	★	商品データ（product）の各商品について、利益率が30%となる新たな単価を求めよ。今回は、1円未満を切り上げること。そして結果を10件表示させ、利益率がおよそ30%付近であることを確認せよ。ただし、単価（unit_price）と原価（unit_cost）には欠損が生じていることに注意せよ。
S-068	四則演算	小数の扱い	乗算結果に対して有効桁数以下を切り捨てる	★	商品データ（product）の各商品について、消費税率10%の税込み金額を求めよ。1円未満の端数は切り捨てとし、結果を10件表示せよ。ただし、単価（unit_price）には欠損が生じていることに注意せよ。
S-069	四則演算	集計結果の演算	集計結果から割合を計算する	★★	レシート明細データ（receipt）と商品データ（product）を結合し、顧客毎に全商品の売上金額合計と、カテゴリ大区分コード（category_major_cd）が"07"（瓶詰缶詰）の売上金額合計を計算の上、両者の比率を求めよ。抽出対象はカテゴリ大区分コード"07"（瓶詰缶詰）の売上実績がある顧客のみとし、結果を10件表示せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
S-070	日付型の計算	経過日数の計算	2つの日付から経過日数を計算する	★★	レシート明細データ (receipt) の売上日 (sales_ymd) に対し、顧客データ (customer) の会員登録日 (application_date) からの経過日数を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。
S-071	日付型の計算	経過日数の計算	2つの日付から経過月数を計算する	★★	レシート明細データ (receipt) の売上日 (sales_ymd) に対し、顧客データ (customer) の会員登録日 (application_date) からの経過月数を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。1ヶ月未満は切り捨てること。
S-072	日付型の計算	経過日数の計算	2つの日付から経過年数を計算する	★★	レシート明細データ (receipt) の売上日 (customer) に対し、顧客データ (customer) の会員登録日 (application_date) からの経過年数を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。1年未満は切り捨てること。
S-073	日付型の計算	経過時間の計算	・2つの日付から経過時間をエポック秒で計算する	★★	レシート明細データ (receipt) の売上日 (sales_ymd) に対し、顧客データ (customer) の会員登録日 (application_date) からのエポック秒による経過時間を計算し、顧客ID (customer_id)、売上日、会員登録日とともに10件表示せよ (なお、sales_ymdは数値、application_dateは文字列でデータを保持している点に注意)。なお、時間情報は保有していないため各日付は0時0分0秒を表すものとする。
S-074	日付型の計算	経過時間の計算	月曜日からの経過日数を計算する	★★	レシート明細データ (receipt) の売上日 (sales_ymd) に対し、当該週の月曜日からの経過日数を計算し、売上日、直前の月曜日付とともに10件表示せよ (sales_ymdは数値でデータを保持している点に注意)。
S-075	サンプリング	ランダム	ランダムサンプリングを行う(単純無作為抽出)	★	顧客データ (customer) からランダムに1%のデータを抽出し、先頭から10件表示せよ。
S-076	サンプリング	層化	カテゴリの割合に応じたサンプリングを行う(層化抽出)	★	顧客データ (customer) から性別コード (gender_cd) の割合に基づきランダムに10%のデータを層化抽出し、性別コードごとに件数を集計せよ。
S-077	外れ値・異常値	外れ値除外	統計的に外れ値を除外する (3σ外の除外)	★	レシート明細データ (receipt) の売上金額を顧客単位に合計し、合計した売上金額の外れ値を抽出せよ。なお、外れ値は売上金額合計を対数化したうえで平均と標準偏差を計算し、その平均から3σを超えて離れたものとする (自然対数と常用対数のどちらでも可)。結果は10件表示せよ。
S-078	外れ値・異常値	外れ値除外	統計的に外れ値を除外する (IQR1.5倍)	★★	レシート明細データ (receipt) の売上金額 (amount) を顧客単位に合計し、合計した売上金額の外れ値を抽出せよ。ただし、顧客IDが"Z"から始まるものは非会員を表すため、除外して計算すること。なお、ここでは外れ値を第1四分位と第3四分位の差であるIQRを用いて、「第1四分位数-1.5×IQR」を下回るもの、または「第3四分位数+1.5×IQR」を超えるものとする。結果は10件表示せよ。
S-079	欠損値	欠損列状況確認	欠損値がある列を確認する	★	商品データ (product) の各項目に対し、欠損数を確認せよ。
S-080	欠損値	欠損レコード削除	欠損値があるレコードを削除する	★	商品データ (product) のいずれかの項目に欠損が発生しているレコードを全て削除した新たな商品データを作成せよ。なお、削除前後の件数を表示させ、079で確認した件数だけ減少していることも確認すること。
S-081	欠損値	数値補完 (平均値)	平均値を用いて欠損値を補完する	★	単価 (unit_price) と原価 (unit_cost) の欠損値について、それぞれの平均値で補完した新たな商品データを作成せよ。なお、平均値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
S-082	欠損値	数値補完 (中央値)	中央値を用いて欠損値を補完する	★	単価 (unit_price) と原価 (unit_cost) の欠損値について、それぞれの中央値で補完した新たな商品データを作成せよ。なお、中央値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
S-083	欠損値	数値補完 (カテゴリごとの中央値)	カテゴリごとに算出した中央値で欠損値を補完する	★★	単価 (unit_price) と原価 (unit_cost) の欠損値について、各商品のカテゴリ小区分コード (category_small_cd) ごとに算出した中央値で補完した新たな商品データを作成せよ。なお、中央値については1円未満を丸めること (四捨五入または偶数への丸めで良い)。補完実施後、各項目について欠損が生じていないことも確認すること。
S-084	除算エラー対応	0で代替	分母が欠損や0の場合でも除算結果データを作成する	★★	顧客データ (customer) の全顧客に対して全期間の売上金額に占める2019年売上金額の割合を計算し、新たなデータを作成せよ。ただし、売上実績がない場合は0として扱うこと。そして計算した割合が0超のものを抽出し、結果を10件表示せよ。また、作成したデータに欠損が存在しないことを確認せよ。
S-085	座標データ	ジオコード	郵便番号からジオコードに変換する	★	顧客データ (customer) の全顧客に対し、郵便番号 (postal_cd) を用いてジオコードデータ (geocode) を紐付け、新たな顧客データを作成せよ。ただし、1つの郵便番号 (postal_cd) に複数の経度 (longitude)、緯度 (latitude) 情報が紐づく場合は、経度 (longitude)、緯度 (latitude) の平均値を算出して使用すること。また、作成結果を確認するために結果を10件表示せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
S-086	座標データ	ジオコード	経度緯度から距離を計算する	★★★★	085で作成した緯度経度つき顧客データに対し、会員申込店舗コード（application_store_cd）をキーに店舗データ（store）と結合せよ。そして申込み店舗の緯度（latitude）・経度情報（longitude）と顧客住所（address）の緯度・経度を用いて申込み店舗と顧客住所の距離（単位：km）を求め、顧客ID（customer_id）、顧客住所（address）、店舗住所（address）とともに表示せよ。計算式は以下の簡易式で良いものとするが、その他精度の高い方式を利用したライブラリを利用してもかまわない。結果は10件表示せよ。  緯度（ラジアン）： $\phi$ 経度（ラジアン）： $\lambda$ 距離 $L=6371*\arccos(\sin\phi1*\sin\phi2+\cos\phi1*\cos\phi2*\cos(\lambda1-\lambda2))$
S-087	名寄せ	完全一致	PK以外の項目を利用した名寄せを行う	★★	顧客データ（customer）では、異なる店舗での申込みなどにより同一顧客が複数登録されている。名前（customer_name）と郵便番号（postal_cd）が同じ顧客は同一顧客とみなして1顧客1レコードとなるように名寄せした名寄顧客データを作成し、顧客データの件数、名寄顧客データの件数、重複数を算出せよ。ただし、同一顧客に対しては売上金額合計が最も高いものを残し、売上金額合計が同一もしくは売上実績がない顧客については顧客ID（customer_id）の番号が小さいものを残すこととする。
S-088	名寄せ	変換データ作成	名寄変換データを作成する	★★	087で作成したデータを元に、顧客データに統合名寄IDを付与したデータを作成せよ。ただし、統合名寄IDは以下の仕様で付与するものとする。  重複していない顧客：顧客ID（customer_id）を設定 重複している顧客：前設問で抽出したレコードの顧客IDを設定 顧客IDのユニーク件数と、統合名寄IDのユニーク件数の差を確認すること。
S-089	データ分割	レコードデータ	ホールドアウト法によるデータの分割を行う	★	売上実績がある顧客を、予測モデル構築のため学習用データとテスト用データに分割したい。それぞれ8:2の割合でランダムにデータを分割せよ。
S-090	データ分割	時系列データ	時系列データを分割する	★★	レシート明細データ（receipt）は2017年1月1日～2019年10月31日までのデータを有している。売上金額（amount）を月次で集計し、学習用に12ヶ月、テスト用に6ヶ月の時系列モデル構築用データを3セット作成せよ。
S-091	不均衡データ	アンダーサンプリング	アンダーサンプリングにより不均衡データを調整する	★	顧客データ（customer）の各顧客に対し、売上実績がある顧客数と売上実績がない顧客数が1:1となるようにアンダーサンプリングで抽出せよ。
S-092	正規化・非正規化	正規化	第三正規形に正規化する	★	顧客データ（customer）の性別について、第三正規形へと正規化せよ。
S-093	正規化・非正規化	非正規化	第三正規形から非正規化する	★	商品データ（product）では各カテゴリのコード値だけを保有し、カテゴリ名は保有していない。カテゴリデータ（category）と組み合わせて非正規化し、カテゴリ名を保有した新たな商品データを作成せよ。
S-094	正規化・非正規化	CSV出力（ヘッダ有り、コード変換なし）	ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する①	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。 - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること（COPYコマンドの権限は付与済み）。 - 出力先：/tmp/data ※"/tmp/data"を指定することでJupyterの"/work/data"と共有されるようになっている。
S-095	ファイル入出力	CSV出力（ヘッダ有り、UTF-8 -> SJIS）	ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する②	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。 - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：CP932  PostgreSQLではShift_JISを指定することでCP932相当となる。ファイル出力先のパスは以下のようにすること（COPYコマンドの権限は付与済み）。 - 出力先：/tmp/data ※"/tmp/data"を指定することでJupyterの"/work/data"と共有されるようになっている。
S-096	ファイル入出力	CSV出力（ヘッダ無し、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを作成する③	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。 - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：無し - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること（COPYコマンドの権限は付与済み）。 - 出力先：/tmp/data ※"/tmp/data"を指定することでJupyterの"/work/data"と共有されるようになっている。

No	大分類	小分類	設問主旨	難易度	設問内容
S-097	ファイル入出力	CSV入力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを読み込む①	★	094で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8
S-098	ファイル入出力	CSV入力（ヘッダ無し、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、CSVファイルを読み込む②	★	096で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：CSV（カンマ区切り） - ヘッダ有無：無し - 文字エンコーディング：UTF-8
S-099	ファイル入出力	TSV出力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、TSVファイルを作成する	★	093で作成したカテゴリ名付き商品データを以下の仕様でファイル出力せよ。 - ファイル形式：TSV（タブ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8  ファイル出力先のパスは以下のようにすること（COPYコマンドの権限は付与済み）。 - 出力先：/tmp/data ※"/tmp/data"を指定することでJupyterの"/work/data"と共有されるようになっている。
S-100	ファイル入出力	TSV入力（ヘッダ有り、コード変換なし）	・ヘッダ有無と文字エンコーディングを指定し、TSVファイルを読み込む	★	099で作成した以下形式のファイルを読み込み、データを3件を表示させて正しく取り込まれていることを確認せよ。  - ファイル形式：TSV（タブ区切り） - ヘッダ有無：有り - 文字エンコーディング：UTF-8