データサイエンス100本ノック（構造化データ加工編） Python版

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	★	レシート明細のデータフレーム（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	★	レシート明細のデータフレーム（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、10件表示させよ。
P-003	列に対する操作	列名変更	・指定列の列名を変更する	★	レシート明細のデータフレーム（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、10件表示させよ。ただし、sales_ymdはsales_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(=,>,<)	★	レシート明細のデータフレーム（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	★	レシート明細のデータフレーム（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" - 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	★	レシート明細データフレーム「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	★	レシート明細のデータフレーム（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" - 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（!=）	★	レシート明細のデータフレーム（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" - 商品コード（product_cd）が"P071401019"以外
P-009	行に対する操作	補集合	・AND、ORで抽出される結果の補集合を取得する	★	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	★	店舗データフレーム（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件だけ表示せよ。
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	★	顧客データフレーム（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件だけ表示せよ。
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	★	店舗データフレーム（df_store）から横浜市の店舗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	★★	顧客データフレーム（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件だけ表示せよ。
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	★★	顧客データフレーム（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件だけ表示せよ。
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	★★	顧客データフレーム（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件だけ表示せよ。
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	★★	店舗データフレーム（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。

No	大分類	小分類	設問主旨	難易度	設問内容
P-017	ソート	並び替え	・データを昇順に並べる	★	顧客データフレーム（df_customer）を生年月日（birth_day）で高齢順にソートし、先頭10件を全項目表示せよ。
P-018	ソート	並び替え	・データを降順に並べる	★	顧客データフレーム（df_customer）を生年月日（birth_day）で若い順にソートし、先頭10件を全項目表示せよ。
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フレーム（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10件を抽出せよ。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フレーム（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10件を抽出せよ。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フレーム（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フレーム（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フレーム（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フレーム（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上日（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フレーム（df_receipt）に対し、顧客ID（customer_id）ごとに最も古い売上日（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フレーム（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上日（sales_ymd）と古い売上日を求め、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フレーム（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フレーム（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フレーム（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フレーム（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標本分散を計算し、降順でTOP5を表示せよ。
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フレーム（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標本標準偏差を計算し、降順でTOP5を表示せよ。
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フレーム（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	・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フレーム（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フレーム（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"から始まるの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フレーム（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せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。なお、データは10件だけ表示させれば良い。
P-036	結合	単一キー	・一項目の結合キーを使ってテーブルを結合する	★	レシート明細データフレーム（df_receipt）と店舗データフレーム（df_store）を内部結合し、レシート明細データフレームの全項目と店舗データフレームの店舗名（store_name）を10件表示させよ。
P-037	結合	複数キー	・複数の結合キーを使ってテーブルを結合する	★	商品データフレーム（df_product）とカテゴリデータフレーム（df_category）を内部結合し、商品データフレームの全項目とカテゴリデータフレームの小区分名（category_small_name）を10件表示させよ。

No	大分類	小分類	設問主旨	難易度	設問内容
P-038	結合	左外部結合	・左外部結合でデータを残す	★	顧客データフレーム（df_customer）とレシート明細データフレーム（df_receipt）から、各顧客ごとの売上金額合計を求めよ。ただし、買い物の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。なお、結果は10件だけ表示させれば良い。
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フレーム（df_receipt）から売上日数の多い顧客の上位20件と、売上金額合計の多い顧客の上位20件を抽出し、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ると何件のデータとなるか調査したい。店舗（df_store）と商品（df_product）を直積した件数を計算せよ。
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フレーム（df_receipt）の売上金額（amount）を日付（sales_ymd）ごとに集計し、前日からの売上金額増減を計算せよ。なお、計算結果は10件表示すればよい。
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フレーム（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、1日前、2日前、3日前のデータを結合せよ。結果は10件表示すればよい。
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フレーム（df_receipt）と顧客データフレーム（df_customer）を結合し、性別（gender）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フレーム（df_sales_summary）を作成せよ。性別は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	★	前設問で作成した売上サマリデータフレーム（df_sales_summary）は性別の売上を横持ちさせたものであった。このデータフレーム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	★	顧客データフレーム（df_customer）の生年月日（birth_day）は日付型（Date）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）とともに抽出せよ。データは10件を抽出すれば良い。
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	★	顧客データフレーム（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（dateやdatetime）に変換し、顧客ID（customer_id）とともに抽出せよ。データは10件を抽出すれば良い。
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フレーム（df_receipt）の売上日（sales_ymd）はYYYYMMDD形式の数値型でデータを保有している。これを日付型（dateやdatetime）に変換し、レシート番号(receipt_no)、レシートサブ番号（receipt_sub_no）とともに抽出せよ。データは10件を抽出すれば良い。
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（dateやdatetime）に変換し、レシート番号(receipt_no)、レシートサブ番号（receipt_sub_no）とともに抽出せよ。データは10件を抽出すれば良い。
P-049	データ変換	日付要素の取り出し	・日付データから特定の年だけ取り出す	★	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）を日付型（timestamp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）とともに抽出せよ。データは10件を抽出すれば良い。
P-050	データ変換	日付要素の取り出し	・日付データから特定の月だけ取り出す	★	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）を日付型（timestamp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）とともに抽出せよ。なお、"月"は0埋め2桁で取り出すこと。データは10件を抽出すれば良い。
P-051	データ変換	日付要素の取り出し	・日付データから特定の日だけ取り出す	★	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）を日付型（timestamp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）とともに抽出せよ。なお、"日"は0埋め2桁で取り出すこと。データは10件を抽出すれば良い。
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フレーム（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2000円以下を0、2000円超を1に2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"から始まるのものは非会員を表すため、除外して計算すること。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★★	顧客データフレーム（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２値化せよ。さらにレシート明細データフレーム（df_receipt）と結合し、全期間において買い物実績のある顧客数を、作成した2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	★★	顧客データデータフレーム（df_customer）の住所（address）は、埼玉県、千葉県、東京都、神奈川県の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、住所とともに抽出せよ。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。結果は10件表示させれば良い。
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	★★	レシート明細データフレーム（df_receipt）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、売上金額と合計ともに表示せよ。カテゴリ値は上から順に1～4とする。結果は10件表示させれば良い。 - 最小値以上第一四分位未満 - 第一四分位以上第二四分位未満 - 第二四分位以上第三四分位未満 - 第三四分位以上
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	★★	顧客データフレーム（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）とともに抽出せよ。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。先頭10件を表示させればよい。
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	★	前問題の抽出結果と性別（gender）を組み合わせ、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せよ。組み合わせを表すカテゴリの値は任意とする。先頭10件を表示させればよい。
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）	★	顧客データフレーム（df_customer）の性別コード（gender_cd）をダミー変数化し、顧客ID（customer_id）とともに抽出せよ。結果は10件表示させれば良い。
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フレーム（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表示せよ。標準化に使用する標準偏差は、不偏標準偏差と標本標準偏差のどちらでも良いものとする。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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フレーム（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を最小値0、最大値1に正規化して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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フレーム（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を常用対数化（底=10）して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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フレーム（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を自然対数化(底=e)して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
P-063	四則演算	四則演算	・数値を引き算する	★	商品データフレーム（df_product）の単価（unit_price）と原価（unit_cost）から、各商品の利益額を算出せよ。結果は10件表示させれば良い。
P-064	四則演算	四則演算	・数値を割り算する	★	商品データフレーム（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NULLが存在することに注意せよ。
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	★	商品データフレーム（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NULLが存在することに注意せよ。
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	★	商品データフレーム（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円未満を四捨五入すること（0.5については偶数方向の丸めで良い）。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NULLが存在することに注意せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★	商品データフレーム（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NULLが存在することに注意せよ。
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	★	商品データフレーム（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、結果は10件表示すれば良い。ただし、単価（unit_price）にはNULLが存在することに注意せよ。
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フレーム（df_receipt）と商品データフレーム（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（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"07"（瓶詰缶詰）の購入実績がある顧客のみとし、結果は10件表示させればよい。
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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い（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）。
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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い（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）。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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い。（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）。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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い（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	・月曜日からの経過日数を計算する	★★	レシート明細データフレーム（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、当該週の月曜日付とともに表示せよ。結果は10件表示させれば良い（なお、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)	★	顧客データフレーム（df_customer）からランダムに1%のデータを抽出し、先頭から10件データを抽出せよ。
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)	★	顧客データフレーム（df_customer）から性別（gender_cd）の割合に基づきランダムに10%のデータを層化抽出データし、性別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フレーム（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"から始まるのものは非会員を表すため、除外して計算すること。なお、ここでは外れ値を平均から3σ以上離れたものとする。結果は10件表示させれば良い。
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フレーム（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"から始まるの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一四分位と第三四分位の差であるIQRを用いて、「第一四分位数-1.5×IQR」よりも下回るもの、または「第三四分位数+1.5×IQR」を超えるものとする。結果は10件表示させれば良い。
P-079	欠損値	欠損列状況確認	・欠損値がある列を確認する	★	商品データフレーム（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	★	商品データフレーム（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df_product_1を作成せよ。なお、削除前後の件数を表示させ、前設問で確認した件数だけ減少していることも確認すること。

No	大分類	小分類	設問主旨	難易度	設問内容
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df_product_2を作成せよ。なお、平均値について1円未満は四捨五入とし、0.5については偶数寄せでかまわない。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df_product_3を作成せよ。なお、中央値について1円未満は四捨五入とし、0.5については偶数寄せでかまわない。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
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、各商品の小区分（category_small_cd）ごとに算出した中央値で補完した新たなdf_product_4を作成せよ。なお、中央値について1円未満は四捨五入とし、0.5については偶数寄せでかまわない。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NULLや0の場合でも除算結果データを作成する	★★	顧客データフレーム（df_customer）の全顧客に対し、全期間の売上金額に占める2019年売上金額の割合を計算せよ。ただし、販売実績のない場合は0として扱うこと。そして計算した割合が0超のものを抽出せよ。 結果は10件表示させれば良い。また、作成したデータにNAやNANが存在しないことを確認せよ。
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	★	顧客データフレーム（df_customer）の全顧客に対し、郵便番号（postal_cd）を用いて経度緯度変換用データフレーム（df_geocode）を紐付け、新たなdf_customer_1を作成せよ。ただし、複数紐づく場合は経度（longitude）、緯度（latitude）それぞれ平均を算出すること。
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	★★★	前設問で作成した緯度経度つき顧客データフレーム（df_customer_1）に対し、申込み店舗コード（application_store_cd）をキーに店舗データフレーム（df_store）と結合せよ。そして申込み店舗の緯度（latitude）・経度情報（longitude)と顧客の緯度・経度を用いて距離（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件表示すれば良い。
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	★★	顧客データフレーム（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、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フレーム（df_customer_u）を作成せよ。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すものとし、売上金額合計が同一もしくは売上実績の無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	★★	前設問で作成したデータを元に、顧客データフレームに統合名寄IDを付与したデータフレーム（df_customer_n）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
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	★	売上実績のある顧客に対し、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フレーム（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	★	顧客データフレーム（df_customer）の各顧客に対し、売上実績のある顧客数と売上実績のない顧客数が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	・非正規化データから第三正規化データを作成する	★	顧客データフレーム（df_customer）では、性別に関する情報が非正規化の状態で保持されている。これを第三正規化せよ。
P-093	正規化・非正規化	非正規化	・第三正規化されたデータから非正規化データを作成する	★	商品データフレーム（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。カテゴリデータフレーム（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フレームを作成せよ。
P-094	正規化・非正規化	CSV出力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はCSV（カンマ区切り） - ヘッダ有り - 文字コードはUTF-8

No	大分類	小分類	設問主旨	難易度	設問内容
P-095	ファイル入出力	CSV出力（ヘッダ有り、UTF-8 -> SJIS）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はCSV（カンマ区切り） - ヘッダ有り - 文字コードはCP932
P-096	ファイル入出力	CSV出力（ヘッダ無し、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はCSV（カンマ区切り） - ヘッダ無し - 文字コードはUTF-8
P-097	ファイル入出力	CSV入力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、データフレームを作成せよ。また、先頭10件を表示させ、正しくとりまれていることを確認せよ。 - ファイル形式はCSV（カンマ区切り） - ヘッダ有り - 文字コードは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）	・文字コードとヘッダ有無を指定しながらCSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、データフレームを作成せよ。また、先頭10件を表示させ、正しくとりまれていることを確認せよ。 - ファイル形式はCSV（カンマ区切り） - ヘッダ無し - 文字コードは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）	・文字コードとヘッダ有無を指定しながらTSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はTSV（タブ区切り） - ヘッダ有り - 文字コードはUTF-8
P-100	ファイル入出力	TSV入力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらTSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、データフレームを作成せよ。また、先頭10件を表示させ、正しくとりまれていることを確認せよ。 - ファイル形式はTSV（タブ区切り） - ヘッダ有り - 文字コードは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	★	レシート明細のデータフレーム（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	★	レシート明細のデータフレーム（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、10件表示させよ。
R-003	列に対する操作	列名変更	・指定列の列名を変更する	★	レシート明細のデータフレーム（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、10件表示させよ。ただし、sales_ymdはsales_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(=,>,<)	★	レシート明細のデータフレーム（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	★	レシート明細のデータフレーム（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" - 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	★	レシート明細データフレーム「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	★	レシート明細のデータフレーム（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" - 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（!=）	★	レシート明細のデータフレーム（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" - 商品コード（product_cd）が"P071401019"以外
R-009	行に対する操作	補集合	・AND、ORで抽出される結果の補集合を取得する	★	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)')
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	★	店舗データフレーム（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件だけ表示せよ。
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	★	顧客データフレーム（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件だけ表示せよ。
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	★	店舗データフレーム（df_store）から横浜市の店舗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	★★	顧客データフレーム（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件だけ表示せよ。
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	★★	顧客データフレーム（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件だけ表示せよ。
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	★★	顧客データフレーム（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件だけ表示せよ。
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	★★	店舗データフレーム（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。

No	大分類	小分類	設問主旨	難易度	設問内容
R-017	ソート	並び替え	・データを昇順に並べる	★	顧客データフレーム（df_customer）を生年月日（birth_day）で高齢順にソートし、先頭10件を全項目表示せよ。
R-018	ソート	並び替え	・データを降順に並べる	★	顧客データフレーム（df_customer）を生年月日（birth_day）で若い順にソートし、先頭10件を全項目表示せよ。
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フレーム（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10件を抽出せよ。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フレーム（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10件を抽出せよ。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フレーム（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フレーム（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フレーム（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フレーム（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上日（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フレーム（df_receipt）に対し、顧客ID（customer_id）ごとに最も古い売上日（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フレーム（df_receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上日（sales_ymd）と古い売上日を求め、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フレーム（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フレーム（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フレーム（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フレーム（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標本分散を計算し、降順でTOP5を表示せよ。
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フレーム（df_receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標本標準偏差を計算し、降順でTOP5を表示せよ。
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フレーム（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	・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フレーム（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フレーム（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フレーム（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せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。なお、データは10件だけ表示させれば良い。
R-036	結合	単一キー	・一項目の結合キーを使ってテーブルを結合する	★	レシート明細データフレーム（df_receipt）と店舗データフレーム（df_store）を内部結合し、レシート明細データフレームの全項目と店舗データフレームの店舗名（store_name）を10件表示させよ。
R-037	結合	複数キー	・複数の結合キーを使ってテーブルを結合する	★	商品データフレーム（df_product）とカテゴリデータフレーム（df_category）を内部結合し、商品データフレームの全項目とカテゴリデータフレームの小区分名（category_small_name）を10件表示させよ。

No	大分類	小分類	設問主旨	難易度	設問内容
R-038	結合	左外部結合	・左外部結合でデータを残す	★	顧客データフレーム（df_customer）とレシート明細データフレーム（df_receipt）から、各顧客ごとの売上金額合計を求めよ。ただし、買い物の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。なお、結果は10件だけ表示させれば良い。
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フレーム（df_receipt）から売上日数の多い顧客の上位20件と、売上金額合計の多い顧客の上位20件を抽出し、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ると何件のデータとなるか調査したい。店舗（df_store）と商品（df_product）を直積した件数を計算せよ。
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フレーム（df_receipt）の売上金額（amount）を日付（sales_ymd）ごとに集計し、前日からの売上金額増減を計算せよ。なお、計算結果は10件表示すればよい。
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フレーム（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、１日前、２日前、３日前のデータを結合せよ。結果は10件表示すればよい。
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フレーム（df_receipt）と顧客データフレーム（df_customer）を結合し、性別（gender）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フレーム（df_sales_summary）を作成せよ。性別は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	★	前設問で作成した売上サマリデータフレーム（df_sales_summary）は性別の売上を横持ちさせたものであった。このデータフレーム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	★	顧客データフレーム（df_customer）の生年月日（birth_day）は日付型（Date）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）とともに抽出せよ。データは10件を抽出すれば良い。
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	★	顧客データフレーム（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（dateやdatetime）に変換し、顧客ID（customer_id）とともに抽出せよ。データは10件を抽出すれば良い。
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フレーム（df_receipt）の売上日（sales_ymd）はYYYYMMDD形式の数値型でデータを保有している。これを日付型（dateやdatetime）に変換し、レシート番号(receipt_no)、レシートサブ番号（receipt_sub_no）とともに抽出せよ。データは10件を抽出すれば良い。
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（dateやdatetime）に変換し、レシート番号(receipt_no)、レシートサブ番号（receipt_sub_no）とともに抽出せよ。データは10件を抽出すれば良い。
R-049	データ変換	日付要素の取り出し	・日付データから特定の年だけ取り出す	★	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）を日付型（timestamp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）とともに抽出せよ。データは10件を抽出すれば良い。
R-050	データ変換	日付要素の取り出し	・日付データから特定の月だけ取り出す	★	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）を日付型（timestamp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）とともに抽出せよ。なお、"月"は0埋め2桁で取り出すこと。データは10件を抽出すれば良い。
R-051	データ変換	日付要素の取り出し	・日付データから特定の日だけ取り出す	★	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）を日付型（timestamp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）とともに抽出せよ。なお、"日"は0埋め2桁で取り出すこと。データは10件を抽出すれば良い。
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フレーム（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2000円以下を0、2000円超を1に2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"から始まるのものは非会員を表すため、除外して計算すること。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★★	顧客データフレーム（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２値化せよ。さらにレシート明細データフレーム（df_receipt）と結合し、全期間において買い物実績のある顧客数を、作成した2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	★	顧客データデータフレーム（df_customer）の住所（address）は、埼玉県、千葉県、東京都、神奈川県の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、住所とともに抽出せよ。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。結果は10件表示させれば良い。
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	★	レシート明細データフレーム（df_receipt）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、売上金額と合計ともに表示せよ。カテゴリ値は上から順に1～4とする。結果は10件表示させれば良い。 - 最小値以上第一四分位未満 - 第一四分位以上第二四分位未満 - 第二四分位以上第三四分位未満 - 第三四分位以上
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	★	顧客データフレーム（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）とともに抽出せよ。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。先頭10件を表示させればよい。
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	★	前問題の抽出結果と性別（gender）を組み合わせ、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せよ。組み合わせを表すカテゴリの値は任意とする。先頭10件を表示させればよい。
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）	★★	顧客データフレーム（df_customer）の性別コード（gender_cd）をダミー変数化し、顧客ID（customer_id）とともに抽出せよ。結果は10件表示させれば良い。
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フレーム（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表示せよ。標準化に使用する標準偏差は、不偏標準偏差と標本標準偏差のどちらでも良いものとする。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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フレーム（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を最小値0、最大値1に正規化して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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フレーム（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を常用対数化（底=10）して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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フレーム（df_receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を自然対数化(底=e）して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
R-063	四則演算	四則演算	・数値を引き算する	★	商品データフレーム（df_product）の単価（unit_price）と原価（unit_cost）から、各商品の利益額を算出せよ。結果は10件表示させれば良い。
R-064	四則演算	四則演算	・数値を割り算する	★	商品データフレーム（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NULLが存在することに注意せよ。
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	★	商品データフレーム（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NULLが存在することに注意せよ。
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	★	商品データフレーム（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円未満を四捨五入すること（0.5については偶数方向の丸めで良い）。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NULLが存在することに注意せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★	商品データフレーム（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NULLが存在することに注意せよ。
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	★	商品データフレーム（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、結果は10件表示すれば良い。ただし、単価（unit_price）にはNULLが存在することに注意せよ。
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フレーム（df_receipt）と商品データフレーム（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（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"07"（瓶詰缶詰）の購入実績がある顧客のみとし、結果は10件表示させればよい。
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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い（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）。
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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い（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）。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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い。（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）。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フレーム（df_receipt）の売上日（sales_ymd）に対し、顧客データフレーム（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表示せよ。結果は10件表示させれば良い（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	・月曜日からの経過日数を計算する	★★★	レシート明細データフレーム（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、当該週の月曜日付とともに表示せよ。結果は10件表示させれば良い（なお、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)	★	顧客データフレーム（df_customer）からランダムに1%のデータを抽出し、先頭から10件データを抽出せよ。
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)	★	顧客データフレーム（df_customer）から性別（gender_cd）の割合に基づきランダムに10%のデータを層化抽出データし、性別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フレーム（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"から始まるのものは非会員を表すため、除外して計算すること。なお、ここでは外れ値を平均から3σ以上離れたものとする。結果は10件表示させれば良い。
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フレーム（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"から始まるの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一四分位と第三四分位の差であるIQRを用いて、「第一四分位数-1.5×IQR」よりも下回るもの、または「第三四分位数+1.5×IQR」を超えるものとする。結果は10件表示させれば良い。
R-079	欠損値	欠損列状況確認	・欠損値がある列を確認する	★	商品データフレーム（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	★	商品データフレーム（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df_product_1を作成せよ。なお、削除前後の件数を表示させ、前設問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df_product_2を作成せよ。なお、平均値について1円未満は四捨五入とし、0.5については偶数寄せでかまわない。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。

No	大分類	小分類	設問主旨	難易度	設問内容
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df_product_3を作成せよ。なお、中央値について1円未満は四捨五入とし、0.5については偶数寄せでかまわない。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
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、各商品の小区分（category_small_cd）ごとに算出した中央値で補完した新たなdf_product_4を作成せよ。なお、中央値について1円未満は四捨五入とし、0.5については偶数寄せでかまわない。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NULLや0の場合でも除算結果データを作成する	★★	顧客データフレーム（df_customer）の全顧客に対し、全期間の売上金額に占める2019年売上金額の割合を計算せよ。ただし、販売実績のない場合は0として扱うこと。そして計算した割合が0超のものを抽出せよ。 結果は10件表示させれば良い。また、作成したデータにNAやNANが存在しないことを確認せよ。
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	★	顧客データフレーム（df_customer）の全顧客に対し、郵便番号（postal_cd）を用いて経度緯度変換用データフレーム（df_geocode）を紐付け、新たなdf_customer_1を作成せよ。ただし、複数紐づく場合は経度（longitude）、緯度（latitude）それぞれ平均を算出すること。
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	★★★	前設問で作成した緯度経度つき顧客データフレーム（df_customer_1）に対し、申込み店舗コード（application_store_cd）をキーに店舗データフレーム（df_store）と結合せよ。そして申込み店舗の緯度（latitude）・経度情報（longitude)と顧客の緯度・経度を用いて距離（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件表示すれば良い。
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	★★	顧客データフレーム（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、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フレーム（df_customer_u）を作成せよ。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すものとし、売上金額合計が同一もしくは売上実績の無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	★★	前設問で作成したデータを元に、顧客データフレームに統合名寄IDを付与したデータフレーム（df_customer_n）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
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	★	売上実績のある顧客に対し、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フレーム（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	★★	顧客データフレーム（df_customer）の各顧客に対し、売上実績のある顧客数と売上実績のない顧客数が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	・非正規化データから第三正規化データを作成する	★	顧客データフレーム（df_customer）では、性別に関する情報が非正規化の状態で保持されている。これを第三正規化せよ。
R-093	正規化・非正規化	非正規化	・第三正規化されたデータから非正規化データを作成する	★	商品データフレーム（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。カテゴリデータフレーム（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フレームを作成せよ。
R-094	正規化・非正規化	CSV出力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はCSV（カンマ区切り） - ヘッダ有り - 文字コードはUTF-8
R-095	ファイル入出力	CSV出力（ヘッダ有り、UTF-8 -> SJIS）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はCSV（カンマ区切り） - ヘッダ有り - 文字コードはCP932

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）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はCSV（カンマ区切り） - ヘッダ無し - 文字コードはUTF-8
R-097	ファイル入出力	CSV入力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、データフレームを作成せよ。また、先頭10件を表示させ、正しくとりまれていることを確認せよ。 - ファイル形式はCSV（カンマ区切り） - ヘッダ有り - 文字コードは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）	・文字コードとヘッダ有無を指定しながらCSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、データフレームを作成せよ。また、先頭10件を表示させ、正しくとりまれていることを確認せよ。 - ファイル形式はCSV（カンマ区切り） - ヘッダ無し - 文字コードは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）	・文字コードとヘッダ有無を指定しながらTSVファイルを作成する	★	先に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。なお、出力先のパスはdata配下とする。 - ファイル形式はTSV（タブ区切り） - ヘッダ有り - 文字コードはUTF-8
R-100	ファイル入出力	TSV入力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらTSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、データフレームを作成せよ。また、先頭10件を表示させ、正しくとりまれていることを確認せよ。 - ファイル形式はTSV（タブ区切り） - ヘッダ有り - 文字コードはUTF-8

データサイエンス100本ノック（構造化データ加工編） SQL版

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	★	レシート明細テーブル（receipt）から全項目を10件抽出し、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	★	レシート明細のテーブル（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、10件表示させよ。
S-003	列に対する操作	列名変更	・指定列の列名を変更する	★	レシート明細のテーブル（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、10件表示させよ。ただし、sales_ymdはsales_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(=,>,<)	★	レシート明細のテーブル（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	★	レシート明細のテーブル（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" 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	★	レシート明細テーブル（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	★	レシート明細のテーブル（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" - 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（!=）	★	レシート明細テーブル（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" - 商品コード（product_cd）が"P071401019"以外
S-009	行に対する操作	補集合	・AND、ORで抽出される結果の補集合を取得する	★	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	★	店舗テーブル（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件だけ表示せよ。
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	★	顧客テーブル（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件だけ表示せよ。
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	★	店舗テーブル（store）から横浜市の店舗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	★★	顧客テーブル（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件だけ表示せよ。
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	★★	顧客テーブル（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件だけ表示せよ。
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	★★	顧客テーブル（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件だけ表示せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★★	店舗テーブル（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	★	顧客テーブル（customer）を生年月日（birth_day）で高齢順にソートし、先頭10件を全項目表示せよ。
S-018	ソート	並び替え	・データを降順に並べる	★	顧客テーブル（customer）を生年月日（birth_day）で若い順にソートし、先頭10件を全項目表示せよ。
S-019	ソート	順位	・順位付けする（同一順位あり）	★★	レシート明細テーブル（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10件を抽出せよ。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）	★★	レシート明細テーブル（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10件を抽出せよ。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	★	レシート明細テーブル（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	★	レシート明細テーブル（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	★	レシート明細テーブル（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	★	レシート明細テーブル（receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上日（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	★	レシート明細テーブル（receipt）に対し、顧客ID（customer_id）ごとに最も古い売上日（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	★	レシート明細テーブル（receipt）に対し、顧客ID（customer_id）ごとに最も新しい売上日（sales_ymd）と古い売上日を求め、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	★	レシート明細テーブル（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	★	レシート明細テーブル（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	★★	レシート明細テーブル（receipt）に対し、店舗コード（store_cd）ごとに商品コードの最頻値を求めよ。
S-030	集計	統計量	・対象データの分散を求める	★	レシート明細テーブル（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標本分散を計算し、降順にTOP5を表示せよ。
S-031	集計	統計量	・対象データの標準偏差を求める	★	レシート明細テーブル（receipt）に対し、店舗コード（store_cd）ごとに売上金額（amount）の標本標準偏差を計算し、降順にTOP5を表示せよ。
S-032	集計	統計量	・データのパーセンタイル値を求める	★	レシート明細テーブル（receipt）に対し、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	・集計結果に対する条件指定で絞り込む	★	レシート明細テーブル（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	★	レシート明細テーブル（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"から始まるの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	★★	レシート明細テーブル（receipt）に対し、顧客ID（customer_id）ごとに販売金額（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せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。なお、データは10件だけ表示させれば良い。
S-036	結合	単一キー	・一項目の結合キーを使ってテーブルを結合する	★	レシート明細テーブル（receipt）と店舗テーブル（store）を内部結合し、レシート明細テーブルの全項目と店舗テーブルの店舗名（store_name）を10件表示させよ。

No	大分類	小分類	設問主旨	難易度	設問内容
S-037	結合	複数キー	・複数の結合キーを使ってテーブルを結合する	★	商品テーブル（product）とカテゴリテーブル（category）を内部結合し、商品テーブルの全項目とカテゴリテーブルの小区分名（category_small_name）を10件表示させよ。
S-038	結合	左外部結合	・左外部結合でデータを残す	★	顧客テーブル（customer）とレシート明細テーブル（receipt）から、各顧客ごとの売上金額合計を求めよ。ただし、買い物の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。なお、結果は10件だけ表示させれば良い。
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	★	レシート明細テーブル（receipt）から売上日数の多い顧客の上位20件と、売上金額合計の多い顧客の上位20件を抽出し、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ると何件のデータとなるか調査したい。店舗（store）と商品（product）を直積した件数を計算せよ。
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	★★	レシート明細テーブル（receipt）の売上金額（amount）を日付（sales_ymd）ごとに集計し、前日からの売上金額増減を計算せよ。なお、計算結果は10件表示すればよい。
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	★★	レシート明細テーブル（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、１日前、２日前、３日前のデータを結合せよ。結果は10件表示すればよい。
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	★	レシート明細テーブル（receipt）と顧客テーブル（customer）を結合し、性別（gender）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テーブル（sales_summary）を作成せよ。性別は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	★	前設問で作成した売上サマリテーブル（sales_summary）は性別の売上を横持ちさせたものであった。このテーブル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	★	顧客テーブル（customer）の生年月日（birth_day）は日付型（Date）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）とともに抽出せよ。データは10件を抽出すれば良い。
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	★	顧客テーブル（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（dateやdatetime）に変換し、顧客ID（customer_id）とともに抽出せよ。データは10件を抽出すれば良い。
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	★	レシート明細テーブル（receipt）の売上日（sales_ymd）はYYYYMMDD形式の数値型でデータを保有している。これを日付型（dateやdatetime）に変換し、レシート番号(receipt_no)、レシートサブ番号（receipt_sub_no）とともに抽出せよ。データは10件を抽出すれば良い。
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	★	レシート明細テーブル（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（timestamp型）に変換し、レシート番号(receipt_no)、レシートサブ番号（receipt_sub_no）とともに抽出せよ。データは10件を抽出すれば良い。
S-049	データ変換	日付要素の取り出し	・日付データから特定の年だけ取り出す	★	レシート明細テーブル（receipt）の販売エポック秒（sales_epoch）を日付型（timestamp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）とともに抽出せよ。データは10件を抽出すれば良い。
S-050	データ変換	日付要素の取り出し	・日付データから特定の月だけ取り出す	★	レシート明細テーブル（receipt）の売上エポック秒（sales_epoch）を日付型（timestamp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）とともに抽出せよ。なお、"月"は0埋め2桁で取り出すこと。データは10件を抽出すれば良い。
S-051	データ変換	日付要素の取り出し	・日付データから特定の日だけ取り出す	★	レシート明細テーブル（receipt）の売上エポック秒（sales_epoch）を日付型（timestamp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）とともに抽出せよ。なお、"日"は0埋め2桁で取り出すこと。データは10件を抽出すれば良い。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★	レシート明細テーブル（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2000円以下を0、2000円超を1に2値化し、顧客ID、合計金額とともに10件表示せよ。ただし、顧客IDが"Z"から始まるの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	★	顧客テーブル（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２値化せよ。さらにレシート明細テーブル（receipt）と結合し、全期間において買い物実績のある顧客数を、作成した2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	★	顧客テーブル（customer）の住所（address）は、埼玉県、千葉県、東京都、神奈川県の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、住所とともに抽出せよ。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。結果は10件表示させれば良い。
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	★	レシート明細テーブル（receipt）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表示せよ。カテゴリ値は上から順に1～4とする。結果は10件表示させれば良い。 - 最小値以上第一四分位未満 - 第一四分位以上第二四分位未満 - 第二四分位以上第三四分位未満 - 第三四分位以上
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	★	顧客テーブル（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）とともに抽出せよ。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。先頭10件を表示させればよい。
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	★	前問題の抽出結果と性別（gender）を組み合わせ、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せよ。組み合わせを表すカテゴリの値は任意とする。先頭10件を表示させればよい。
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）	★★	顧客テーブル（customer）の性別コード（gender_cd）をダミー変数化し、顧客ID（customer_id）とともに抽出せよ。結果は10件表示させれば良い。
S-059	数値変換	正規化（z-score）	・平均0、分散1に変換する	★	レシート明細テーブル（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を平均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表示せよ。標準化に使用する標準偏差は、不偏標準偏差と標本標準偏差のどちらでも良いものとする。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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	★	レシート明細テーブル（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を最小値0、最大値1に正規化して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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）	★	レシート明細テーブル（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を常用対数化（底=10）して顧客ID、売上金額合計とともに表示せよ。ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること。結果は10件表示させれば良い。
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）	★	レシート明細テーブル（receipt）の売上金額（amount）を顧客ID（customer_id）ごとに合計し、合計した売上金額を自然対数化(底=e）して顧客ID、売上金額合計とともに表示せよ（ただし、顧客IDが"Z"から始まるのものは非会員を表すため、除外して計算すること）。結果は10件表示させれば良い。
S-063	四則演算	四則演算	・数値を引き算する	★	商品テーブル（product）の単価（unit_price）と原価（unit_cost）から、各商品の利益額を算出せよ。結果は10件表示させれば良い。
S-064	四則演算	四則演算	・数値を割り算する	★	商品テーブル（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NULLが存在することに注意せよ。
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	★	商品テーブル（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NULLが存在することに注意せよ。

No	大分類	小分類	設問主旨	難易度	設問内容
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	★	商品テーブル（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円未満を四捨五入すること。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NULLが存在することに注意せよ。
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	★	商品テーブル（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NULLが存在することに注意せよ。
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	★	商品テーブル（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、結果は10件表示すれば良い。ただし、単価（unit_price）にはNULLが存在することに注意せよ。
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	★★	レシート明細テーブル（receipt）と商品テーブル（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（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"07"（瓶詰缶詰）の購入実績がある顧客のみとし、結果は10件表示させればよい。
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	★★	レシート明細テーブル（receipt）の売上日（sales_ymd）に対し、顧客テーブル（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表示せよ。結果は10件表示させれば良い（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）。
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	★★	レシート明細テーブル（receipt）の売上日（sales_ymd）に対し、顧客テーブル（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表示せよ。結果は10件表示させれば良い（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）。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	★★	レシート明細テーブル（receipt）の売上日（sales_ymd）に対し、顧客テーブル（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表示せよ。結果は10件表示させれば良い（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）。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	★★	レシート明細テーブル（receipt）の売上日（sales_ymd）に対し、顧客テーブル（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表示せよ。結果は10件表示させれば良い（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	★★	レシート明細テーブル（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、当該週の月曜日付とともに表示せよ。結果は10件表示させれば良い（なお、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)	★	顧客テーブル（customer）からランダムに1%のデータを抽出し、先頭から10件データを抽出せよ。
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)	★★★	顧客テーブル（customer）から性別（gender_cd）の割合に基づきランダムに10%のデータを層化抽出データし、性別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）	★	レシート明細テーブル（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"から始まるのものは非会員を表すため、除外して計算すること。なお、ここでは外れ値を平均から3σ以上離れたものとする。結果は10件表示させれば良い。
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）	★★	レシート明細テーブル（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"から始まるの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一四分位と第三四分位の差であるIQRを用いて、「第一四分位数-1.5×IQR」よりも下回るもの、または「第三四分位数+1.5×IQR」を超えるものとする。結果は10件表示させれば良い。
S-079	欠損値	欠損列状況確認	・欠損値がある列を確認する	★	商品テーブル（product）の各項目に対し、欠損数を確認せよ。

データサイエンス100本ノック（構造化データ加工編） SQL版

No	大分類	小分類	設問主旨	難易度	設問内容
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	★	商品テーブル（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product_1を作成せよ。なお、削除前後の件数を表示させ、前設問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product_2を作成せよ。なお、平均値について1円未満は四捨五入とする。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product_3を作成せよ。なお、中央値について1円未満は四捨五入とする。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、各商品の小区分（category_small_cd）ごとに算出した中央値で補完した新たなproduct_4を作成せよ。なお、中央値について1円未満は四捨五入とする。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NULLや0の場合でも除算結果データを作成する	★★	顧客テーブル（customer）の全顧客に対し、全期間の売上金額に占める2019年売上金額の割合を計算せよ。ただし、販売実績のない場合は0として扱うこと。そして計算した割合が0超のものを抽出せよ。 結果は10件表示させれば良い。
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	★	顧客テーブル（customer）の全顧客に対し、郵便番号（postal_cd）を用いて経度緯度変換用テーブル（geocode）を紐付け、新たなcustomer_1を作成せよ。ただし、複数紐づく場合は経度（longitude）、緯度（latitude）それぞれ平均を算出すること。
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	★★★	前設問で作成した緯度経度つき顧客テーブル（customer_1）に対し、申込み店舗コード（application_store_cd）をキーに店舗テーブル（store）と結合せよ。そして申込み店舗の緯度（latitude）・経度情報（longitude)と顧客の緯度・経度を用いて距離（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件表示すれば良い。
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	★★	顧客テーブル（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、1顧客1レコードとなるように名寄せした名寄顧客テーブル（customer_u）を作成せよ。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すものとし、売上金額合計が同一もしくは売上実績の無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	★★	前設問で作成したデータを元に、顧客テーブルに統合名寄IDを付与したテーブル（customer_n）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
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	★	売上実績のある顧客に対し、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分割し、テーブルを作成せよ。
S-090	データ分割	時系列データ	・時系列データを分割する	★★★	レシート明細テーブル（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ヶ月のモデル構築用データを3テーブルとしてセット作成せよ。データの持ち方は自由とする。
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	★★★	顧客テーブル（customer）の各顧客に対し、売上実績のある顧客数と売上実績のない顧客数が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	・非正規化データから第三正規化データを作成する	★	顧客テーブル（customer）では、性別に関する情報が非正規化の状態で保持されている。これを第三正規化せよ。
S-093	正規化・非正規化	非正規化	・第三正規化されたデータから非正規化データを作成する	★	商品テーブル（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。カテゴリテーブル（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テーブルを作成せよ。

データサイエンス100本ノック（構造化データ加工編） SQL版

No	大分類	小分類	設問主旨	難易度	設問内容
S-094	正規化・非正規化	CSV出力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。出力先のパスは'/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。なお、COPYコマンドの権限は付与済みである。 ・ファイル形式はCSV（カンマ区切り） ・ヘッダ有り ・文字コードはUTF-8
S-095	ファイル入出力	CSV出力（ヘッダ有り、UTF-8 -> SJIS）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。出力先のパスは'/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。なお、COPYコマンドの権限は付与済みである。 ・ファイル形式はCSV（カンマ区切り） ・ヘッダ有り ・文字コードはSJIS
S-096	ファイル入出力	CSV出力（ヘッダ無し、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを作成する	★	先に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。出力先のパスは'/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。なお、COPYコマンドの権限は付与済みである。 ・ファイル形式はCSV（カンマ区切り） ・ヘッダ無し ・文字コードはUTF-8
S-097	ファイル入出力	CSV入力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらCSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、テーブルを作成せよ。また、先頭3件を表示させ、正しくとりまれていることを確認せよ。 ・ファイル形式はCSV（カンマ区切り） ・ヘッダ有り ・文字コードは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）	・文字コードとヘッダ有無を指定しながらCSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、テーブルを作成せよ。また、先頭3件を表示させ、正しくとりまれていることを確認せよ。 ・ファイル形式はCSV（カンマ区切り） ・ヘッダ無し ・文字コードは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）	・文字コードとヘッダ有無を指定しながらTSVファイルを作成する	★	先に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。出力先のパスは'/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。なお、COPYコマンドの権限は付与済みである。 ・ファイル形式はTSV（タブ区切り） ・ヘッダ有り ・文字コードはUTF-8
S-100	ファイル入出力	TSV入力（ヘッダ有り、コード変換なし）	・文字コードとヘッダ有無を指定しながらTSVファイルを読み込む	★	先に作成した以下形式のファイルを読み込み、テーブルを作成せよ。また、先頭3件を表示させ、正しくとりまれていることを確認せよ。 ・ファイル形式はTSV（タブ区切り） ・ヘッダ有り ・文字コードはUTF-8