```sql
--Here I will check for columns that contain NULL values, which can affect your
  analysis and may need to be addressed.
SELECT
    COUNT(*) AS total_rows,
    SUM(CASE WHEN credit_policy IS NULL THEN 1 ELSE 0 END) AS missing_credit_policy,
    SUM(CASE WHEN purpose IS NULL THEN 1 ELSE 0 END) AS missing_purpose,
    SUM(CASE WHEN int_rate IS NULL THEN 1 ELSE 0 END) AS missing_int_rate,
    SUM(CASE WHEN installment IS NULL THEN 1 ELSE 0 END) AS missing_installment,
    SUM(CASE WHEN log_annual_inc IS NULL THEN 1 ELSE 0 END) AS missing_log_annual_inc,
    SUM(CASE WHEN dti IS NULL THEN 1 ELSE 0 END) AS missing_dti,
    SUM(CASE WHEN fico IS NULL THEN 1 ELSE 0 END) AS missing_fico,
    SUM(CASE WHEN revol_bal IS NULL THEN 1 ELSE 0 END) AS missing_revol_bal,
    SUM(CASE WHEN revol_util IS NULL THEN 1 ELSE 0 END) AS missing_revol_util,
    SUM(CASE WHEN inq_last_6mths IS NULL THEN 1 ELSE 0 END) AS missing_inq_last_6mths,
    SUM(CASE WHEN delinq_2yrs IS NULL THEN 1 ELSE 0 END) AS missing_delinq_2yrs,
    SUM(CASE WHEN pub_rec IS NULL THEN 1 ELSE 0 END) AS missing_pub_rec,
    SUM(CASE WHEN not_fully_paid IS NULL THEN 1 ELSE 0 END) AS missing_not_fully_paid
FROM
    dbo.loandata;

--Here, I will look for data points that don't make sense, such as a FICO score that's
   too low to be realistic: FICO scores below a reasonable threshold (500) or Interest
   rates that are negative or too high.
SELECT
    COUNT(*) AS anomalous_fico_count
FROM
    dbo.loandata
WHERE
    fico < 500;
SELECT
    COUNT(*) AS anomalous_interest_count
FROM
    dbo.loandata
WHERE
    int_rate < 0 OR int_rate > 1;

--Here, I want to ensure that the data is consistent. I want to make sure that loans
  marked as not fully paid have a non-zero balance.
SELECT
    COUNT(*) AS inconsistent_records
FROM
    dbo.loandata
WHERE
    not_fully_paid = 1 AND revol_bal = 0;


--Assuming I have determined that the inconsistency is indeed an error and the
  revol_bal should be greater than 0, I will set it to a NULL.
UPDATE dbo.loandata
SET revol_bal = NULL
WHERE not_fully_paid = 1 AND revol_bal = 0;
```

```sql
--Duplicate records can skew my results, so I wanted to identify and remove any
  duplicates from my dataset.
SELECT
  credit_policy, purpose, int_rate, installment, log_annual_inc, dti, fico,
    days_with_cr_line,
  revol_bal, revol_util, inq_last_6mths, delinq_2yrs, pub_rec, not_fully_paid, COUNT
    (*) AS DuplicateCount
FROM
  dbo.loandata
GROUP BY
  credit_policy, purpose, int_rate, installment, log_annual_inc, dti, fico,
    days_with_cr_line,
  revol_bal, revol_util, inq_last_6mths, delinq_2yrs, pub_rec, not_fully_paid
HAVING
  COUNT(*) > 1;
```