/\* 1. Create a library \*/

libname hospital 'C:\\SASProjects\\HospitalData';

/\* 2. Create dataset using DATALINES \*/

data hospital.patients;

infile datalines dsd truncover;

input PatientID :8. Name :$20. Gender :$1. DOB :yymmdd10. AdmissionDate :yymmdd10.

DischargeDate :yymmdd10. Diagnosis :$20. BillAmount :8. RoomType :$10. Insurance :$3.;

format DOB AdmissionDate DischargeDate yymmdd10.;

datalines;

101,John Smith,M,1980-05-10,2023-01-15,2023-01-25,Pneumonia,45000,General,Yes

102,Linda Brown,F,1992-08-21,2023-02-10,2023-02-12,,32000,Private,

103,Robert Green,M,,2023-03-05,2023-03-15,Diabetes,28000,General,No

104,Susan White,F,1985-12-03,2023-04-01,2023-04-10,Hypertension,56000,ICU,Yes

105,,F,1990-06-12,2023-05-20,,Asthma,.,General,No

106,Mark Taylor,M,1978-11-30,2023-06-18,2023-06-28,Hypertension,60000,Private,

107,Emily Davis,F,1995-04-22,2023-07-10,2023-07-14,,39000,ICU,Yes

;

run;

/\* 3. Rename and reorder columns \*/

data hospital.patients\_renamed;

retain PatientID Name Gender DateOfBirth AdmissionDate DischargeDate Diagnosis BillAmount RoomType Insurance;

set hospital.patients(rename=(DOB=DateOfBirth));

run;

/\* 4. Add new column using SET \*/

data hospital.patients\_age;

set hospital.patients\_renamed;

if DateOfBirth ne . then Age = int((today() - DateOfBirth)/365.25);

run;

/\* 5. Create a new table from existing \*/

data hospital.icu\_patients;

set hospital.patients\_age;

where RoomType = "ICU";

run;

/\* 6. Add fixed, conditional, and calculated columns \*/

data hospital.patients\_extended;

set hospital.patients\_age;

HospitalBranch = "Central";

if BillAmount > 50000 then RiskLevel = "High";

else if 30000 <= BillAmount <= 50000 then RiskLevel = "Medium";

else if BillAmount < 30000 then RiskLevel = "Low";

if AdmissionDate ne . and DischargeDate ne . then StayDays = DischargeDate - AdmissionDate;

run;

/\* 7. Date-related functions \*/

data hospital.patients\_dateparts;

set hospital.patients\_extended;

BirthYear = year(DateOfBirth);

AdmissionMonth = month(AdmissionDate);

run;

/\* 8. Date formatting \*/

proc datasets library=hospital nolist;

modify patients\_dateparts;

format AdmissionDate ddmmyy10. DateOfBirth monyy7.;

quit;

/\* 9. INFORMAT FORMAT for date \*/

data hospital.patients\_format;

infile datalines dsd truncover;

input PatientID :8. Name :$20. Gender :$1. DOB :yymmdd10. AdmissionDate :yymmdd10.

DischargeDate :yymmdd10. Diagnosis :$20. BillAmount :8. RoomType :$10. Insurance :$3.;

informat DOB AdmissionDate DischargeDate yymmdd10.;

format DOB AdmissionDate DischargeDate yymmdd10.;

datalines;

;

run;

/\* 10. Concatenation of columns \*/

data hospital.patients\_concat;

set hospital.patients\_dateparts;

PatientSummary = catx(' - ', Name, Gender, Diagnosis);

run;

/\* 11. Sorting \*/

proc sort data=hospital.patients\_concat out=hospital.patients\_sorted;

by descending BillAmount;

run;

/\* 12. WHERE filtering \*/

data hospital.patients\_missing\_diag;

set hospital.patients\_concat;

where Diagnosis = '';

run;

/\* 13. IF statement filtering \*/

data hospital.high\_icu;

set hospital.patients\_concat;

if BillAmount > 40000 and RoomType = "ICU";

run;

/\* 14. Complex filtering with KEEP \*/

data hospital.patients\_keep;

set hospital.patients\_concat(keep=Name Diagnosis BillAmount);

run;

/\* 15. DROP variables \*/

data hospital.patients\_drop;

set hospital.patients\_concat(drop=RoomType);

run;

/\* 16. Export to CSV, XLSX, TXT \*/

proc export data=hospital.patients\_concat outfile="C:\\SASProjects\\patients.csv" dbms=csv replace;

run;

proc export data=hospital.patients\_concat outfile="C:\\SASProjects\\patients.xlsx" dbms=xlsx replace;

run;

proc export data=hospital.patients\_concat outfile="C:\\SASProjects\\patients.txt" dbms=tab replace;

run;

/\* 17. Already saved in custom library (hospital) \*/

/\* 18. Import CSV \*/

proc import datafile="C:\\SASProjects\\patients.csv" out=hospital.patients\_import dbms=csv replace;

getnames=yes;

run;

/\* 19. Handling missing values \*/

data hospital.patients\_missing\_flag;

set hospital.patients\_concat;

if Diagnosis = '' then MissingDiagnosis = 1; else MissingDiagnosis = 0;

run;

proc freq data=hospital.patients\_missing\_flag;

tables MissingDiagnosis;

run;

proc means data=hospital.patients\_concat n nmiss;

run;

/\* 20. Remove duplicates \*/

proc sort data=hospital.patients\_concat nodupkey out=hospital.patients\_nodup;

by PatientID;

run;

/\* 21. Handle outliers \*/

proc univariate data=hospital.patients\_concat;

var BillAmount;

run;

data hospital.outliers\_removed;

set hospital.patients\_concat;

if BillAmount <= 60000;

run;

/\* 22. Standardizing text \*/

data hospital.standardized\_diag;

set hospital.patients\_concat;

Diagnosis\_up = upcase(Diagnosis);

Diagnosis\_proper = propcase(Diagnosis);

run;

/\* 23. PROC FORMAT \*/

proc format;

value $genderfmt 'M'='Male' 'F'='Female';

run;

data hospital.patients\_genderfmt;

set hospital.patients\_concat;

format Gender $genderfmt.;

run;

/\* 24. PROC MEANS / FREQ \*/

proc means data=hospital.patients\_concat mean maxdec=2;

var BillAmount;

class Diagnosis;

run;

proc freq data=hospital.patients\_concat;

tables Diagnosis Gender;

run;

/\* 25. PROC RANK \*/

proc rank data=hospital.patients\_concat out=hospital.patients\_rank ties=low;

var BillAmount;

ranks Bill\_Rank;

run;

/\* 26. Create second dataset \*/

data hospital.insurance\_info;

input PatientID CoverageAmount;

datalines;

101 30000

102 25000

103 0

104 50000

105 10000

107 35000

;

run;

/\* 27. Merging datasets \*/

proc sort data=hospital.patients\_concat; by PatientID; run;

proc sort data=hospital.insurance\_info; by PatientID; run;

data hospital.patients\_merged;

merge hospital.patients\_concat(in=a) hospital.insurance\_info(in=b);

by PatientID;

run;