\* Importing data for this assignment;

**proc** **import** datafile= "Z:\Downloads\CaseStudy\_HR.csv"

dbms=csv

out=work.HRDATA

replace;

**run**;

/\* Contents of imported data \*/

**proc** **contents** data=HRDATA;

**run**;

/\*Transforming Categorical variables\*/

**data** HRDATA\_Transformed;

set HRDATA;

/\* Binary Coding \*/

if Attrition = 'Yes' then Attrition\_Num = **1**; else Attrition\_Num = **0**;

if OverTime = 'Yes' then OverTime\_Num = **1**; else OverTime\_Num = **0**;

/\* Ordinal Coding \*/

if BusinessTravel = 'Non-Travel' then BusinessTravel\_Num = **0**;

else if BusinessTravel = 'Travel\_Rarely' then BusinessTravel\_Num = **1**;

else if BusinessTravel = 'Travel\_Frequently' then BusinessTravel\_Num = **2**;

if Gender = 'Male' then Gender\_Num = **1**; else Gender\_Num = **0**;

if MaritalStatus = 'Single' then MaritalStatus\_Num = **0**;

else if MaritalStatus = 'Divorced' then MaritalStatus\_Num = **1**;

else if MaritalStatus = 'Married' then MaritalStatus\_Num = **2**;

/\* Create Dummy Variables for JobRole \*/

if JobRole = 'Sales Executive' then JobRole\_SalesExec = **1**; else JobRole\_SalesExec = **0**;

if JobRole = 'Research Scientist' then JobRole\_ResearchSci = **1**; else JobRole\_ResearchSci = **0**;

if JobRole = 'Laboratory Technician' then JobRole\_LabTech = **1**; else JobRole\_LabTech = **0**;

if JobRole = 'Manufacturing Director' then JobRole\_ManufDir = **1**; else JobRole\_ManufDir = **0**;

if JobRole = 'Healthcare Representative' then JobRole\_HealthRep = **1**; else JobRole\_HealthRep = **0**;

if JobRole = 'Manager' then JobRole\_Manager = **1**; else JobRole\_Manager = **0**;

if JobRole = 'Sales Representative' then JobRole\_SalesRep = **1**; else JobRole\_SalesRep = **0**;

if JobRole = 'Research Director' then JobRole\_ResearchDir = **1**; else JobRole\_ResearchDir = **0**;

if JobRole = 'Human Resources' then JobRole\_HR = **1**; else JobRole\_HR = **0**;

/\* Create Dummy Variables for Department \*/

if Department = 'Sales' then Dept\_Sales = **1**; else Dept\_Sales = **0**;

if Department = 'Research & Development' then Dept\_RD = **1**; else Dept\_RD = **0**;

if Department = 'Human Resources' then Dept\_HR = **1**; else Dept\_HR = **0**;

**run**;

/\* Contents imported data \*/

**PROC** **CONTENTS** data=HRDATA\_Transformed;

**run**;

/\*summary Stats

/\* Summary Statistics \*/

**proc** **means** data=HRDATA\_Transformed n nmiss mean std min max median;

var

age

Gender\_Num

MonthlyIncome

YearsAtCompany

BusinessTravel\_Num

OverTime\_Num

StockOptionLevel

DistanceFromHome

jobsatisfaction

NumCompaniesWorked

Dept\_Sales

YearsSinceLastPromotion

BusinessTravel\_Num

TrainingTimesLastYear

WorkLifeBalance

RelationshipSatisfaction

EnvironmentSatisfaction

; title "Summary Stats";

**run**;

/\*Summary Statistics \*/

/\* Sub Sample Analysis of the Dataset \*/

**proc** **means** data=HRDATA\_Transformed n nmiss mean std min max median;

class Attrition\_Num;

var

age

Gender\_Num

MonthlyIncome

YearsAtCompany

BusinessTravel\_Num

OverTime\_Num

StockOptionLevel

DistanceFromHome

jobsatisfaction

NumCompaniesWorked

YearsSinceLastPromotion

BusinessTravel\_Num

TrainingTimesLastYear

WorkLifeBalance

RelationshipSatisfaction

EnvironmentSatisfaction

;

title "Summary Statistics for Continuous Variables by Attrition Status";

**run**;

/\*\_\_\_\_\_EDA\_\_\_\_\_\_\_\_\_\_\_\*/

/\* Step 1: Calculate Total Count per Department \*/

**proc** **sql**;

create table DeptTotal as

select

Department,

count(\*) as TotalCount

from HRDATA\_Transformed

group by Department;

**quit**;

/\* Step 2: Calculate Attrition Count and Join with Total Count to Get Percentage \*/

**proc** **sql**;

create table DeptAttritionPct as

select

a.Department,

a.Attrition\_Num,

count(\*) as Count,

(count(\*) \* **100.0** / b.TotalCount) as Attrition\_Pct

from HRDATA\_Transformed as a

left join DeptTotal as b

on a.Department = b.Department

group by a.Department, a.Attrition\_Num;

**quit**;

**proc** **sort** data=DeptAttritionPct nodupkey;

by Department Attrition\_Num;

**run**;

**proc** **sgplot** data=DeptAttritionPct;

vbar Department / response=Attrition\_Pct group=Attrition\_Num groupdisplay=cluster datalabel;

xaxis label="Department";

yaxis label="Attrition Percentage" grid;

keylegend / title="Attrition Status";

title "Percentage of Attrition by Department";

**run**;

**proc** **sgplot** data=HRDATA\_Transformed;

scatter x=Age y=MonthlyIncome / group=Attrition\_Num markerattrs=(symbol=circlefilled);

title "Age vs. Monthly Income by Attrition";

xaxis label="Age";

yaxis label="Monthly Income";

**run**;

/\*Avetage Income by Age/Attrition

/\* Step 1: Create Age Buckets \*/

**data** HRDATA\_Bucketed;

set HRDATA\_Transformed;

length Age\_Bucket $**10**;

if **20** <= Age < **25** then Age\_Bucket = '20-25';

else if **25** <= Age < **30** then Age\_Bucket = '25-30';

else if **30** <= Age < **35** then Age\_Bucket = '30-35';

else if **35** <= Age < **40** then Age\_Bucket = '35-40';

else if **40** <= Age < **45** then Age\_Bucket = '40-45';

else if **45** <= Age < **50** then Age\_Bucket = '45-50';

else if **50** <= Age < **55** then Age\_Bucket = '50-55';

else if **55** <= Age <= **60** then Age\_Bucket = '55-60';

**run**;

/\* Step 2: Calculate Average Monthly Income by Age Bucket and Attrition \*/

**proc** **means** data=HRDATA\_Bucketed noprint;

class Age\_Bucket Attrition\_Num;

var MonthlyIncome;

output out=AvgIncomeByBucket mean=Avg\_MonthlyIncome;

**run**;

/\* Step 3: Round Average Monthly Income \*/

**data** AvgIncomeByBucket;

set AvgIncomeByBucket;

Avg\_MonthlyIncome = round(Avg\_MonthlyIncome, **1**); /\* Round to whole number \*/

**run**;

/\* Step 4: Plot Average Monthly Income by Age Bucket and Attrition \*/

**proc** **sgplot** data=AvgIncomeByBucket;

vbar Age\_Bucket / response=Avg\_MonthlyIncome group=Attrition\_Num groupdisplay=cluster datalabel;

xaxis label="Age Bucket";

yaxis label="Average Monthly Income";

title "Average Monthly Income by Age Bucket and Attrition";

keylegend / title="Attrition Status";

**run**;

**proc** **sgplot** data=HRDATA\_Transformed;

vbox MonthlyIncome / category=Attrition\_Num;

title "Monthly Income by Attrition Status";

xaxis label="Attrition Status (0 = Stayed, 1 = Left)";

yaxis label="Monthly Income";

**run**;

**proc** **sgplot** data=HRDATA\_Transformed;

histogram Age / group=Attrition\_Num transparency=**0.5**;

density Age / group=Attrition\_Num type=kernel;

title "Age Distribution by Attrition Status";

xaxis label="Age";

yaxis label="Frequency";

**run**;

**proc** **sgplot** data=HRDATA\_Transformed;

vbar BusinessTravel\_Num / group=Attrition\_Num groupdisplay=cluster;

title "Attrition by Business Travel Frequency";

xaxis label="Business Travel Frequency";

yaxis label="Count";

**run**;

/\* Step 1: Create Age Buckets \*/

**data** HRDATA\_Bucketed;

set HRDATA\_Transformed;

length Age\_Bucket $**10**;

if **20** <= Age < **25** then Age\_Bucket = '<25';

else if **25** <= Age < **30** then Age\_Bucket = '25-30';

else if **30** <= Age < **35** then Age\_Bucket = '30-35';

else if **35** <= Age < **40** then Age\_Bucket = '35-40';

else if **40** <= Age < **45** then Age\_Bucket = '40-45';

else if **45** <= Age < **50** then Age\_Bucket = '45-50';

else if **50** <= Age < **55** then Age\_Bucket = '50-55';

else Age\_Bucket = '>60';

**run**;

/\* Step 2: Calculate Average Job Satisfaction by Age Bucket and Attrition \*/

**proc** **means** data=HRDATA\_Bucketed noprint;

class Age\_Bucket Attrition\_Num;

var JobSatisfaction;

output out=AvgJobSatisfaction mean=Avg\_JobSatisfaction;

**run**;

/\* Step 3: Round Average Job Satisfaction to Two Decimal Places \*/

**data** AvgJobSatisfaction;

set AvgJobSatisfaction;

Avg\_JobSatisfaction = round(Avg\_JobSatisfaction, **0.01**); /\* Round to two decimal places \*/

**run**;

/\* Step 4: Plot Average Job Satisfaction by Age Bucket and Attrition \*/

**proc** **sgplot** data=AvgJobSatisfaction;

vbar Age\_Bucket / response=Avg\_JobSatisfaction group=Attrition\_Num groupdisplay=cluster datalabel datalabelattrs=(size=**10**);

xaxis label="Age Bucket";

yaxis label="Average Job Satisfaction" values=(**1.5** to **4** by **0.2**);

title "Average Job Satisfaction by Age Bucket and Attrition Status";

keylegend / title="Attrition Status";

**run**;

/\*\_\_\_\_Segmentation\_\_\_\_\*/

/\* Segmentation using k-means clustering \*/

**proc** **fastclus** data=HRDATA\_Transformed maxclusters=**3** out=SegmentedData;

var

MonthlyIncome

BusinessTravel\_Num

TrainingTimesLastYear

WorkLifeBalance

RelationshipSatisfaction

EnvironmentSatisfaction

;

**run**;

**proc** **tabulate** data=SegmentedData;

class Cluster;

var

MonthlyIncome

BusinessTravel\_Num

TrainingTimesLastYear

WorkLifeBalance

RelationshipSatisfaction

EnvironmentSatisfaction

;

table Cluster,

(

MonthlyIncome

BusinessTravel\_Num

TrainingTimesLastYear

WorkLifeBalance

RelationshipSatisfaction

EnvironmentSatisfaction) \* (mean std);

title "Segment Profile by Cluster - Mean and Standard Deviation";

**run**;

/\* Viewing the contents of segmented data \*/

**proc** **contents** data=SegmentedData;

**run**;

**proc** **export** data=SegmentedData

outfile="Z:\Downloads\SegmentedData.xlsx"

dbms=xlsx

replace;

**run**;

/\*\_\_\_\_\_\_\_Survival Analysis\_\_\_\_\_\_\_\_\*/

**data** HRDATA\_SURV;

set HRDATA\_Transformed;

Event=Attrition\_Num;

Time = YearsAtCompany;

**run**;

**proc** **lifetest** data=HRDATA\_SURV plots=(s);

time Time \* Event(**0**);

strata Department; /\* Optional: Stratify by Department or any other categorical variable \*/

**run**;

/\* Baseline Cox Proportional Hazards Model \*/

**proc** **phreg** data=SegmentedData;

/\* Categorical Variables \*/

class cluster(ref='2') Gender\_Num(ref='0') BusinessTravel\_Num(ref='0')

OverTime\_Num(ref='0') JobRole(ref='Sales Executive') / param=ref;

/\* Model Statement \*/

model YearsAtCompany\*Attrition\_Num(**0**) =

Gender\_Num Age

NumCompaniesWorked

OverTime\_Num

StockOptionLevel

Dept\_Sales

DistanceFromHome

JobSatisfaction

YearsSinceLastPromotion

cluster/ ties=efron;

title 'Baseline Cox Proportional Hazards Model for Employee Attrition';

**run**;

/\* Kaplan Meier Estimation \*/

ods graphics on;

**proc** **lifetest** data=SegmentedData plots=survival (atrisk=**0**);

time YearsAtCompany\*Attrition\_Num(**0**);

strata cluster; /\* Example of stratifying by employed status, you can stratify by other variables \*/

ods select survivalplot;

title " Survival";

**run**;

/\* Kaplan Meier Estimation \*/

ods graphics on;

**proc** **lifetest** data=SegmentedData method=LT intervals=(**0** TO **36** by **1**) plots=(survival) outsurv=survdata;

time YearsAtCompany\*Attrition\_Num(**0**);

strata cluster;

**run**;

ods graphics on;

/\*survdata\*/

**proc** **sgplot** data=survdata;

title 'Survival Probability Estimates';

series x=YearsAtCompany y=Survival / group=cluster;

YAXIS min=**0.0**;

**run**;

/\* Interaction Cox Proportional Hazards Model \*/

**proc** **phreg** data=SegmentedData;

/\* Categorical Variables \*/

class MaritalStatus\_num(ref='2') cluster(ref='2') Gender\_Num(ref='0')

BusinessTravel\_Num(ref='0') OverTime\_Num(ref='0')

JobRole(ref='Sales Executive') / param=ref;

/\* Model Statement \*/

model YearsAtCompany\*Attrition\_Num(**0**) =

BusinessTravel\_Num\*OverTime\_Num age

StockOptionLevel

StockOptionLevel\*age

age\*jobsatisfaction

NumCompaniesWorked

Dept\_Sales

YearsSinceLastPromotion\*cluster

/ ties=efron;

title 'Baseline Cox Proportional Hazards Model for Employee Attrition';

**run**;