**proc** **print** data=sortedpilotdata split = '+';

var employeeid firstname lastname salary;

label employeeid = 'ID+Number'

FirstName = 'First+Name'

Lastname = 'Last+Name'

Salary = 'Annual+Salary';

format salary dollar12.2;

**run**;

**data** practice.pilotdata;

infile 'c:\users\jdanehower\documents\my sas files\practicefiles\pilot.dat';

input EmployeeID $ **1** - **6**

FirstName $ **7** - **19**

LastName $ **20** - **34**

JobCode $ **35** - **41**

Salary **42** -**47**

Category $ **48** -**50**;

bonus=salary \* **.1**;

if jobcode='PILOT1' then NewSalary = Salary\***1.05**;

else if jobcode='PILOT2' then NewSalary= Salary\***1.07**;

else NewSalary= Salary\***1.09**;

run;

**proc** **means** data=practice.pilotdata;

var salary newsalary;

class category;

where jobcode = 'PILOT2';

**run**;

**proc** **means** data=practice.pilotdata;

var salary newsalary;

class category;

where category = 'DOM';

**run**;

**proc** **means** data=practice.pilotdata;

var salary newsalary;

class category;

where category = 'DOM' and jobcode = 'PILOT2';

**run**;

**proc** **freq** data = practice.pilotdata;

tables category\*jobcode;

**run**;

**Create SAS Data set and Add two new variables**

**data** practice.ratedata;

infile 'c:\users\jdanehower\documents\my sas files\practicefiles\rates.dat';

input Origin $ **1** - **3**

Destination $ **6** - **8**

FlightRange $ **12** - **17**

CargoRate **20** - **23**

PassengerFare **28** -**34**

FlightCategory $ **38** -**40**;

NewCargoRate=CargoRate+**.50**;

If FlightRange='SHORT' then

NewPassengerFare=PassengerFare\***1.08**;

else if FlightRange='MEDIUM' then

NewPassengerFare=PassengerFare\***1.10**;

else if FlightRange='LONG' then

NewPassengerFare=PassengerFare\***1.12**;

run;

Sort Data by variables origin and flightrange, call it rateresorted as opposed to ratedata:

**proc** **sort** data = practice.ratedata

out = practice.rateresorted;

by origin flightrange; (you could also add DESCENDING in the by statement )

**run**;

List out practice.resorted including only the variables origin, destination, flightrange, passengerfare, newpassengerfare, labeling passengerfare to Passenger Fare and Newpassengerfare to New Passenger Fare. Formatting Passengerfare and newpassengerfare to dollars with two decimal places, adding one title and one footnote.

**proc** **print** data = practice.rateresorted label;

var origin destination flightrange passengerfare newpassengerfare;

label passengerfare= 'Passenger Fare' Newpassengerfare = 'New Passenger Fare';

format Passengerfare dollar9.2;

format newpassengerfare dollar9.2;

title1 'List Report of Practice.Ratesorted with Selected Variables';

footnote1 'Created by Josh D';

**run**;

Compare the averages of passengerfare and newpassengerfare for flights with RDU as the destination, separated/ held by which whether the flight is domestic or international. So class tells SAS to separate the means by different flight categories. It separates by multiple variables, it could separate by more, but this only has two variables, domestic and international. Where destination = RDU selects only flights that have that as a destination. So we’re only getting the flights where the destination is RDU.(Where the destination equals the text listed as ‘RDU’)

**proc** **means** data= practice.rateresorted;

var passengerfare newpassengerfare;

where destination = 'RDU';

class flightcategory;

title;

footnote;

**run**;

**This shows how many flights fall into each flightcategory, where flightrange equals short**

**proc** **freq** data=practice.rateresorted;

tables flightcategory;

where flightrange= 'SHORT';

**run**;

This shows a two way table with flightcategory and flightrange

**proc** **freq** data=practice.rateresorted;

tables flightcategory\*flightrange;

**run**;

**Names library Orion**

LIBNAME Orion 'c:\users\jdanehower\documents\my sas files\9.3';

**Shows the contents of the Orion library**

**proc** **contents** data=orion.\_ALL\_ NODS;

**run**;

**Shows the contents of the Orion library**

**proc** **contents** data=orion.\_ALL\_;

**run**;

**Views the descriptor portion of the SAS dataset**

**proc** **contents** data=orion.employee\_addresses;

**run**

**copies the dataset orion.sales to a new dataset called work.subset1**

**data work.subset1;**

**set orion.sales;**

**run;**