

4175

A la suma de las puntuaciones de cada pregunta se le restando -0.5 porque el informe en pdf es demasiado escueto.  
Desktop\LastYear\Data\data.xlsx"

ok (1)

```
)  
PROC IMPORT OUT = e.data  
  DATAFILE = "C:\Users\  
  DBMS = XLSX REPLACE;  
  /* sheet = 'Lleida'; */  
GETNAMES = YES;  
)
```

```
LABEL  
VendorID = 'A code indicating the TPEP provider that provided the record.'  
lpep_pickup_datetime = 'Date Pickup'  
lpep_dropoff_datetime = 'Date Dropoff'  
Unique_Identifier = 'Unique Identifier'  
store_and_fwd_flag = 'This flag indicates whether the trip record was held in  
vehicle memory before sending to the vendor'  
RatecodeID = 'The final rate code in effect at the end of the trip'  
passenger_count = 'The number of passengers in the vehicle'  
Trip_distance = 'Distance in km'  
fare_amount = 'Price in Dollar'  
Tip_amount = 'Tip in Dollar'  
tip_amount = 'Tip amount ¶ This field is automatically populated for credit  
card tips. Cash tips are not included'  
tolls_amount = 'Total amount of all tolls paid in trip.'  
total_amount = 'The total amount charged to passengers. Does not include cash tips.'  
payment_type = 'A numeric code signifying how the passenger paid for the trip.  
1= Credit card; 2= Cash; 3= No charge; 4= Dispute; 5= Unknown; 6= Voided trip';  
  
RUN;  
  
PROC PRINT data = e.data; RUN;
```

ok (1)

```
/* Create a format for the different variables */
```

```
proc format library = e;  
value VendorID  
1 = 'Creative Mobile Technologies,'  
2 = 'VeriFone Inc.';
```

```
value RateCodeID  
1 = 'Standard rate'  
2 = 'JFK'  
3 = 'Newark'  
4 = 'Nassau or Westchester'  
5 = 'Negotiated fare'  
6 = 'Group ride';
```

```
value Payment_type  
1 = 'Credit card'  
2 = 'Cash'  
3 = 'No charge'  
4 = 'Dispute'  
5 = 'Unknown'  
6 = 'Voided trip';
```

```
/* Apply the format */
```

```
options fmtsearch = (e);  
data e.data;  
set e.data;  
format VendorID VendorID. RateCodeID RateCodeID. Payment_type Payment_type.;  
run;
```

3) y 4) Hacen un descriptivo básico, o mejor hacer un proc univariate para detectar observaciones extremas.  
Mejorar la justificación.

```
/* PROC Means for numeric variables */
```

```
proc means data = e.data maxdec = 2 ;  
var passenger_count Trip_distance fare_amount Tip_amount tolls_amount total_amount ;  
run;
```

```
/* In the two variables total_amount and fare_amount with found negative values  
which don't make sense.
```

```
We set negative occurring in those two columns to 0 */
```

```
data e.data;  
set e.data;  
if (total_amount < 0) then total_amount = 0 ;  
if (fare_amount < 0) then fare_amount = 0 ;  
run;
```

↳ mejor poner missing

5)

```
/* Descriptive analysis */  
proc freq data= e.data ;  
table VendorID passenger_count/ nocum;  
title 'Frequency of vendor and passenger amount';  
run;
```

ok (1)

```
proc gchart data = e.data;  
vbar payment_type / discrete inside = freq subgroup = VendorID;  
run;
```

```
PROC SGPLOT data = e.data;  
Scatter x = fare_amount y = tip_amount;  
title 'Finding bad tipppers';  
Run;
```

6) falta (0)

```
PROC UNIVARIATE DATA = e.data noprint;  
HISTOGRAM Trip_distance / NORMAL;  
RUN;
```

```
PROC UNIVARIATE DATA = e.data noprint;  
HISTOGRAM tip_amount / NORMAL;  
RUN;
```

```
title 'tipping by rate code';  
proc gchart data=e.data;  
pie Trip_distance / type=sum sumvar = tolls_amount;  
run;
```

7)

```
/* Covariance Analysis */
```

```
proc iml;
use e.data;
read all var {payment_type fare_amount trip_distance passenger_count} into great_data;

start covariance(x);
```

```
sum=x[+, ] ;
n = nrow(x);
```

```
covariance = (t(x)*x-t(sum)*sum/n)/(n-1);
col = {payment_type fare_amount trip_distance passenger_count};
row = {payment_type fare_amount trip_distance passenger_count};
```

```
print covariance[rowname = row colname = col];
```

```
finish covariance;
run covariance(great_data);
title 'Great Covariance Analysis'
```

```
quit;
```

```
/* Macro for automating the Covariance */
```

```
%macro covariance_macro(frame, vnum1, vnum2, vnum3, vnum4);
```

```
proc iml;
use &frame;
read all var{&vnum1, &vnum2, &vnum3, &vnum4} into mat;
```

```
start covariance(x);
```

```
sum = x[+, ] ;
n = nrow(x);
```

```
covariance = (t(x)*x-t(sum)*sum/n)/(n-1);
```

```
col = {&vnum1, &vnum2, &vnum3, &vnum4};
row = {&vnum1, &vnum2, &vnum3, &vnum4};
```

```
print covariance[rowname = row colname = col];
```

```
finish covariance;
```

```
output = covariance(mat);
print output
quit;
```

```
%mend;
%covariance_macro(e.data, payment_type, fare_amount, trip_distance, passenger_count);
```

es categoría!

0.75 se  
pedía  
cerebrucos!

Aún así, sabe  
deprir función  
de IML

8) 0

9) 0

10)

0.75

Repetición del  
7 en forma de  
macro, pero  
no lo aplique  
a otras situaciones,  
y además repite  
los errores!