/\* GERA OS ARQUIVOS ON THE FLY \*/

/\* THIS WILL BE ITUNES AND BILLBOARD COMBINED \*/

%let VA=%str("various","vários","varios");

**proc** **sql**;

create table all1 as

select

Art,AA,Album,Title,Art as Art\_Orig,AA as AA\_Orig,Title as Title\_Orig

,Location

,length/**60** as Length\_Sec

,substr(location,**1**,findc(location,"\","B")) as Subdir

,lowcase(AA) not in (&VA) as Alb\_by\_Art length=**3**

,calculated Alb\_by\_Art and (%***Greatest\_Hits***(Album)=**0**) as Not\_Great\_Hits length=**3**

,case

when (Covers>**0** and index(lowcase(genre),"nicecover")>**0**) then **0**

else **1** end as Needs\_change length=**3**,covers length=**3**

,case when Covers>**0** then **2** else **0** end as Has\_Cover length=**3**

,Year length=**4**,Genre

from jose.art

where %***both\_pop***(art,title)

and location ne ""

and calculated Has\_Cover=**0**;

/\*and calculated Needs\_change=1\*/

/\*and (calculated Has\_Cover=0 or \*/

/\* calculated Alb\_by\_Art=0 or \*/

/\* calculated Not\_Great\_Hits=0 )\*/

;

**quit**;

/\* eliminate dupes \*/

**proc** **sort** data=all1 nodupkey;

by location;

**run**;

/\* FIRST, CREATE A LIST OF ALBUMS THAT HAVE ART \*/

/\* VARIOUS ArtS TEM QUE SER PADRONIZADO, ELES SAO TODOS A MESMA COISA \*/

**proc** **sql**;

create table Has\_Art as

select

left(trim(lowcase(tranwrd(AA,"""","")))) as AA

,left(trim(lowcase(tranwrd(Album,"""","")))) as Album

,max(Covers) as Has\_Cover\_Db

from all

where AA is not null and Album is not null and Covers>**0**

group by **1**,**2**;

**quit**;

%***Padroniza***(all1,all2);

/\* THE BELOW WILL ALLOW US TO JOIN TO TRACKS02 \*/

**proc** **sort** data=all2;

where Art ne "" and title ne "";

by Art Title;

**run**;

/\* for the not hit, I will match them to itunes \*/

**data** nohit;

merge all2(in=a) jose.tracks5(in=b keep=Art Title);

by Art Title;

if a and not b;

**run**;

/\* finds album for no hits \*/

**proc** **sql**;

create table nohit2(drop=Art Title) as

select a.\*

,b.AA as AA\_Db

,b.album as Album\_Db

,b.Year as Year\_Db

,lowcase(b.AA) not in (&va) as Alb\_by\_Art\_Db length=**3**

from nohit a, all2 b

where a.Art=b.Art and a.Title=b.Title and

b.AA ne "" and b.album ne "" and

a.location ne b.location;

**quit**;

/\* THE TRACKS THAT NEED ALBUM \*/

/\* AQUI APENAS SOUNDTRACK E JP NAO SAO SUBSTITUIDOS POR ArtA ORIGINAL \*/

**data** all3;

merge all2(in=a) jose.tracks5(in=b keep=Art Title Album\_sk);

by Art Title;

if a and b;

drop Art Title;

**run**;

/\* SORT BEFORE JOIN, DO DUPES HERE \*/

**proc** **sort** data=all3;

by Album\_SK;

**run**;

/\* SELECT JUST THE TRACKS THAT MATCH THE ABOVE OBS (DROPS FROM 1.1M TO 834K rows) \*/

**data** all3;

length Year\_Db **4**;

merge all3(in=a) jose.CD\_Data(in=b keep=AA Album Album\_SK Year

rename=(Year=Year\_Db AA=AA\_Db Album=Album\_Db));

by Album\_SK;

if a and b and AA\_Db ne "" and Album\_Db ne "";

/\* PADRONIZA ALBUM Art \*/

select;

when (index(lowcase(AA\_Db),"stratovarius")>**0**);

when (index(lowcase(AA\_Db),"varios")>**0**) AA\_Db="Vários";

when (index(lowcase(AA\_Db),"vári")>**0**) AA\_Db="Vários";

when (index(lowcase(AA\_Db),"vari")>**0** and index(lowcase(Subdir),"brasil\")>**0**) AA\_Db="Vários";

when (lowcase(AA\_Db)="va" and index(lowcase(Subdir),"brasil\")>**0**) AA\_Db="Vários";

when (lowcase(AA\_Db)="v.a." and index(lowcase(Subdir),"brasil\")>**0**) AA\_Db="Vários";

when (index(lowcase(AA\_Db),"vari")>**0**) AA\_Db="Various";

when (lowcase(AA\_Db)="va") AA\_Db="Various";

when (lowcase(AA\_Db)="v.a.") AA\_Db="Various";

otherwise;

end;

AA\_db=left(trim(tranwrd(AA\_db,"""","")));

Album\_Db=left(trim(tranwrd(Album\_db,"""","")));

length Alb\_by\_Art\_Db **3**;

Alb\_by\_Art\_Db=lowcase(AA\_Db) not in (&va);

drop Album\_sk;

**run**;

/\* stack \*/

**data** all4;

set nohit2 all3;

length Live\_Ok **3**;

if (index(lowcase(genre),"live")>**0**)=

(max(%***str\_chk***(Album\_Db,acoust),%***str\_chk***(Album\_Db,live),%***str\_chk***(Album\_Db,unplug),

%***str\_chk***(Album\_Db,acusti),%***str\_chk***(Album\_Db,acústi),%***str\_chk***(Album\_Db,vivo),

%***str\_chk***(Album\_Db,concert),%***str\_chk***(Album\_Db,in conce))>**0**)

then Live\_Ok=**1**;

else Live\_Ok=**0**;

if Live\_Ok;

**run**;

/\* computes statistics \*/

**proc** **means** data=all4 nway noprint missing;

where **1930**<=Year\_Db<=**2017**;

var Year\_Db;

class location Year Alb\_by\_Art\_Db;

output out=percs(drop=\_type\_) min=Mini max=Maxi p1=pc1 p5=pc5 p10=pc10 p25=pc25 median=pc50 n=Freq std=dev;

**run**;

/\* stats \*/

**proc** **freq** data=Percs;

table Alb\_by\_Art\_Db;

**run**;

/\* How many hits per track there are? \*/

/\*

proc means data=Percs min p1 p5 p10 p25 median p75 p90 p95 p99 max;

var Freq;

run;

\*/

/\* SELECT MORE ACCURATE YEAR PER ALBUM (heuristic) \*/

**proc** **sql**;

create table Percs2 as

select location, Year

,max(case when Alb\_by\_Art\_Db then round(pc1,**1**) end) as pc1\_Art

,max(case when Alb\_by\_Art\_Db then round(pc5,**1**) end) as pc5\_Art

,max(case when Alb\_by\_Art\_Db then round(pc10,**1**) end) as pc10\_Art

,max(case when Alb\_by\_Art\_Db then round(pc25,**1**) end) as pc25\_Art

,max(case when Alb\_by\_Art\_Db then round(pc50,**1**) end) as pc50\_Art

/\* VA \*/

,max(case when not Alb\_by\_Art\_Db then round(pc1,**1**) end) as pc1\_VA

,max(case when not Alb\_by\_Art\_Db then round(pc5,**1**) end) as pc5\_VA

,max(case when not Alb\_by\_Art\_Db then round(pc10,**1**) end) as pc10\_VA

,max(case when not Alb\_by\_Art\_Db then round(pc25,**1**) end) as pc25\_VA

,max(case when not Alb\_by\_Art\_Db then round(pc50,**1**) end) as pc50\_VA

,sum(Alb\_by\_Art\_Db\*Freq) as Art

,sum((not Alb\_by\_Art\_Db)\*Freq) as VA

,sum(Freq) as Freq

,round(case

when sum(Alb\_by\_Art\_Db\*Freq)>=**3** and sum(Alb\_by\_Art\_Db\*(Mini=Maxi))>**0** then max(Mini)

when sum(Freq)>=**4** and min(Mini)=max(Maxi) then max(Mini)

when **0.01**\*calculated Art>=**6** then calculated pc1\_Art

when **0.05**\*calculated Art>=**3** then calculated pc5\_Art

when **0.10**\*calculated Art>=**3** then calculated pc10\_Art

when **0.25**\*calculated Art>=**3** then calculated pc25\_Art

when **0.50**\*calculated Art>=**3** then calculated pc50\_Art

when **0.01**\*calculated VA>=**6** then calculated pc1\_VA

when **0.05**\*calculated VA>=**3** then calculated pc5\_VA

when **0.10**\*calculated VA>=**3** then calculated pc10\_VA

when **0.25**\*calculated VA>=**3** then calculated pc25\_VA

when **0.50**\*calculated VA>=**3** then calculated pc50\_VA

when **0.50**\*sum(Freq)>=**4** then (calculated pc50\_Art+calculated pc50\_VA)/**2**

end,**1**) as Year\_Db\_Prob,case

when sum(Alb\_by\_Art\_Db\*Freq)>=**3** and sum(Alb\_by\_Art\_Db\*(Mini=Maxi))>**0** then "no diff-art"

when sum(Freq)>=**4** and min(Mini)=max(Maxi) then "no diff-all"

when **0.01**\*calculated Art>=**6** then "pc1\_Art"

when **0.05**\*calculated Art>=**3** then "pc5\_Art"

when **0.10**\*calculated Art>=**3** then "pc10\_Art"

when **0.25**\*calculated Art>=**3** then "pc25\_Art"

when **0.50**\*calculated Art>=**3** then "pc25\_Art"

when **0.01**\*calculated VA>=**6** then "pc1\_VA"

when **0.05**\*calculated VA>=**3** then "pc5\_VA"

when **0.10**\*calculated VA>=**3** then "pc10\_VA"

when **0.25**\*calculated VA>=**3** then "pc25\_VA"

when **0.50**\*calculated VA>=**3** then "pc50\_VA"

when **0.50**\*sum(Freq)>=**4** then "mix"

end "?" as Pick

from Percs

group by **1**,**2**;

**quit**;

/\* stats \*/

**proc** **freq** data=Percs2;

/\*weight freq;\*/

table Pick / missing ;

**run**;

/\* WHICH ONES HAVE COVER? \*/

**proc** **sql**;

create table all5 as

select Art\_Orig as Art,Title\_Orig as Title,AA\_orig as AA,AA\_Db,AA\_Db as AA\_Db2

,a.Alb\_by\_Art,a.Alb\_by\_Art\_Db

,a.Not\_Great\_Hits,a.Alb\_by\_Art\_Db and (%***Greatest\_Hits***(a.Album\_Db)=**0**) as Not\_Great\_Hits\_Db length=**3**

,a.Has\_Cover length=**3**, coalesce(b.Has\_Cover\_Db,**0**) as Has\_Cover\_Db length=**3**

,coalesce(a.Year,**0**) as Year,coalesce(Year\_Db,**0**) as Year\_Db,coalesce(c.Year\_Db\_Prob,**0**) as Year\_Db\_Prob

,a.\*

from all4 a

left join Has\_Art b on lowcase(a.AA\_Db)=lowcase(b.AA) and lowcase(a.Album\_Db)=lowcase(b.Album)

left join Percs2 c on a.location=c.location;

**quit**;

/\* stats \*/

**proc** **sql**;

select Has\_Cover\_Db

,count(distinct case when has\_cover=**2** then location end) as Has\_cover

,count(distinct case when has\_cover=**0** then location end) as No\_cover

from all5

group by **1**;

**quit**;

/\* padroniza de novo \*/

%***Padroniza***(all5,all6,chave1=Art\_Orig,chave2=Title\_Orig,

chave3=AA\_Orig,chave4=AA\_Db2,perm\_Art\_title=**0**);

/\* Prioritize \*/

/\* the below was wrong, redo!! \*/

**proc** **sql**;

create table all7 as

select count(\*) as N

,Art\_Orig,Title\_Orig,AA\_Orig,AA\_Db2

,max(index(lowcase(Art\_Orig),trim(lowcase(AA\_Db2)))

,index(lowcase(AA\_Db2),trim(lowcase(Art\_Orig)))

,index(lowcase(Title\_Orig),trim(lowcase(AA\_Db2))))>**0** as AA\_Right\_Db length=**3**

,max(index(lowcase(Art\_Orig),trim(lowcase(AA\_orig)))

,index(lowcase(AA\_orig),trim(lowcase(Art\_orig)))

,index(lowcase(Title\_Orig),trim(lowcase(AA\_orig))))>**0** as AA\_Right length=**3**,

/\* score DB \*/

(**10**\*\***8**)\*Has\_Cover\_Db+

(**10**\*\***7**)\*Alb\_By\_Art\_Db+

(**10**\*\***6**)\*(Alb\_By\_Art\_Db and calculated AA\_Right\_Db)+

(**10**\*\***5**)\*(lowcase(Title\_Orig)=lowcase(Album\_Db) and Alb\_By\_Art\_Db)+

(**10**\*\***4**)\*Not\_Great\_Hits\_Db+

(**10**\*\***3**)\*(Year\_Db\_Prob ne **0**)\*max(**0.01**,**1**/(**1**+abs(Year\_Db-Year\_Db\_Prob))) as Score\_Db,

/\* Score \*/

(**10**\*\***8**)\*Has\_Cover+

(**10**\*\***7**)\*Alb\_By\_Art+

(**10**\*\***6**)\*(Alb\_By\_Art and calculated AA\_Right)+

(**10**\*\***5**)\*(lowcase(Title\_Orig)=lowcase(Album) and Alb\_By\_Art)+

(**10**\*\***4**)\*Not\_Great\_Hits+

(**10**\*\***3**)\*(Year\_Db\_Prob ne **0**)\*max(**0.01**,**1**/(**1**+abs(Year-Year\_Db\_Prob))) as Score,\*

from all6 a

where calculated score\_db>calculated score

and not (Has\_Cover=**2** and Has\_Cover\_Db ne **2**)

and not (Has\_cover=**2** and Alb\_By\_Art\_Db=**0**)

group by Location

having score\_db=max(score\_db)

order by Location,Score\_db;

**quit**;

/\* ESSAS EH NO GO\*/

/\*

proc sql;

create table no\_go as

select count(\*) as N,\*

from all6

group by Location

having sum(lowcase(album)=lowcase(album\_db))>0 and sum(AA\_wrong)=0

order by n desc,Location,Score;

quit;

\*/

/\* SE O MAXIMO SCORE EH ATINGIDO PELO ALBUM ATUAL, NAO MUDAR \*/

/\* NAO PODE SER INCLUIDO ACIMA \*/

/\* NOTEI QUE ISSO SE DEVE AO FATO DE POR EX. TER UMA CAPA DISP. EM EXPORTED MAS NAO NA TRACK \*/

**proc** **sql**;

create table all7 as

select \*

from all7

group by Location

having sum(lowcase(album)=lowcase(album\_db))=**0**

order by Location,Score;

**quit**;

/\* ELIMINATES REMAINING DUPES \*/

/\* EXCLUDE ALBUMS THAT DIDN'T CHANGE TO NOT REMOVE THEIR COVERS INCORRECTLY \*/

**proc** **sort** data=all7(rename=(AA\_Db=New\_AA Album\_Db=New\_Album)) out=all8 nodupkey;

where Album="" or AA\_Right\_Db=**0**;

by location;

**run**;

**proc** **freq** data=all7;

table Has\_Cover;

**run**;

/\* COM O ESQUEMA ABAIXO, SO MUDO AS CAPAS QUE FOREM RUINS \*/

/\* CREATE PLAYLIST TO REMOVE ARTWORK \*/

**data** Has\_cover;

set all7;

/\*where Has\_Cover>0;\*/

keep location AA\_db album\_db AA\_orig Album;

**run**;

/\* HERE I HAVE TO REMOVE THE COVERS \*/

**data** has\_cover;

set all7;

if AA\_wrong=**0** and Has\_Cover=**1**;

keep location New\_AA New\_album AA\_orig Album Has\_Cover Cover\_Db\_Prob Has\_Cover\_Db;

**run**;

**proc** **sql**;

select Has\_Cover\_Db,Has\_Cover

,Alb\_By\_Art\_Db,Alb\_By\_Art

,count(\*) as N

from all7

group by **1**,**2**,**3**,**4**;

**quit**;

/\* THE BELOW WILL GENERATE SCRIPT TO ADD THE TRACKS \*/

%let playlist\_name=AAA\_Repop\_Album5;

%***changes\_tags***(&playlist\_name,all7,modify\_album=true);

**proc** **sql**;

create table expo as

select location

from all1

where lowcase(AA) in

(select lowcase(New\_AA) from all7)

and lowcase(AA) not in (&VA) and covers>**0**

group by lowcase(AA),lowcase(album)

having location=max(location);

**quit**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* AFTER THE ABOVE RUNS \*/

/\* EMBED ARTWORK \*/

x "cscript &Drive:\iTunes\scripts\Embed\_Artwork\_v2.vbs";

**proc** **freq** data=all4;

table covers alb\_by\_Art alb\_by\_Art\_db Has\_Cover\_Db Album\_Db\_May\_Have\_Cover;

**run**;

/\* ADICIONA ARQUIVOS PARA QUE CAPAS POSSAM SER COPIADAS DE UM PRA OUTRO \*/

**proc** **sql**;

create table all6 as

select \*

from all

where lowcase(location) not in

(select lowcase(location) from all5);

**quit**;

**data** all7;

retain Will\_change;

set all5(in=a drop=AA Album rename=(New\_AA=AA New\_Album=Album)) all6;

Will\_change=a;

if a then covers=**0**;

**run**;

/\* A PRIMEIRA PROVIDENCIA EH TROCAR ALBUMS QUE NAO ESTAO CORRETOS \*/

/\* ALL POSSIBLE SCENARIOS \*/

%let score\_comp=(10\*\*3)\*(Has\_Cover\_Db>0)+(10\*\*2)\*Alb\_by\_Art\_Db

+10\*Year\_Is\_Ok\_Db+1\*Not\_Great\_Hits\_Db>

(10\*\*3)\*Has\_Cover+(10\*\*2)\*Alb\_by\_Art+10\*Year\_Is\_Ok+1\*Not\_Great\_Hits;

**proc** **sql**;

create table Scenarios as

select Has\_Cover,Has\_Cover\_Db,

Alb\_by\_Art,Alb\_by\_Art\_Db,

Year\_Is\_Ok,Year\_Is\_Ok\_Db,

Not\_Great\_Hits,Not\_Great\_Hits\_Db,

Scenario, count(distinct location) as tracks

from all3

where &score\_comp

group by **1**,**2**,**3**,**4**,**5**,**6**,**7**,**8**,**9**;

**quit**;