Lynn Nguyen

Dr. Wright

STA402-A

13 May 2024

Term Project Update

I. Original Statement of Your Assigned Task

Write a SAS macro that accepts a genre and presents the relationships among the budget, revenue, and size of the cast. Your macro should also allows users to specify a list of genre that they'd like to compare.

Data Description

There are 2 datasets in csv file file `tmdb_5000_credits.csv  tmdb_5000_movies.csv.`

`tmdb_5000_movies.csv : has columns budget, genres, homepage, id, keywords, orignial_language, original_title, overview, popularity, productions_companies,production_countries, release_date, revenue, runtime, spoken_languages, status, tagline, title, vote_average, vote_count with 4803 observations`

`tmdb_5000_credits.csv: has columns movie_id, title, cast, crew.`

II. Progress Report

1. Project Description

I have to merge these 2 movies and credits csv datasets by movie id and movie title. Then, from the merged dataset, I extracted column title, genre, budget, revenue, and cast_size. Finally, write a macro to analyze relationship among budget, revenue, and size of the cast by genre.

2. Explain SAS steps

a. <u>Import Movies and Credits datasets into SAS</u>

- Use proc import to import 2 datasets: movies and credits into SAS
- Specify datafile in proc import and declare the path for getting csv files
- Specify output file to use within in SAS
- Use dbms to SAS the format of the file
- Create dataset name extracted_movies and extracted_credits

b. <u>Merge extracted_movies and extracted_credits</u>

- Use proc sql to merge 2 datasets extracted_movies and extracted_credits by tile and movie_id
- Named the new dataset as merged_data

c. <u>Create genre dataset</u>

- Extract column genre
- Loops through each "{" of genres column and take information as genre_names
- Loops through genre_names and the the information in the 6^{th} position

- Contains the information extracted in column name genre
- Only keep column title and genre
- Use proc sort to sort the genre dataset by title named new dataset as genre_sorted

d. <u>Create cast dataset</u>

- Count the occurrences of "cast_id" in the cast column
- Store the output the column named cast_size

e. <u>Create Budget dataset</u>

- From the merged_data, keep column title and budget
- Use proc sort to sort budget by title and named new dataset as budget_sorted

f. <u>Create Revenue dataset</u>

- From the merged_data, keep column title and revenue
- Use proc sort to sort revenue by title and named new dataset as revenue_sorted

g. <u>Merge Genre_sorted, Cast_sorted, Budget_sorted, and Revenue_sorted</u>

- Merged genre_sorted, cast_sorted, budget_sorted, and revenue_sorted by title named new dataset as combine
- Use proc sort to sort combine by genre named new dataset and combine_sorted
- Remove all the observations that do not have genre

h. <u>Build macro to analyze relationship among revenue, budget, cast_size by genre</u>

- Create a filtered_data by Genre desired
- Use proc means to summary mean of budget, revenue, cast_size by genre desired
- Use proc print to print out the summary stats
- Use proc sgplot to plot the scatter plot between revenue and budget, revenue and cast_size, and budget and cast_size
- It is a user-specified macro with user-specified genre

III. Code and Output related to II.2

a. <u>Import Movies and Credits datasets into SAS</u>

```
%let movies = M:\TermProject\tmdb_5000_movies.csv;

libname myfiles 'M:\TermProject';
/*Import tmdb_5000_movies.csv and create permanent file*/
proc import
    datafile= "&movies"
    out=myfiles.extracted_movies
    replace
    dbms=csv;
    getnames=yes; /* Assumes the first row contains variable names */
    guessingrows=max;
run;

/*Print first 5 observations for myfiles.extracted.movies */
```

```
proc print data=myfiles.extracted_movies (obs=5);
run;

/*Import tmdb_5000_credits.csv from a ZIP folder*/
FILENAME ZIPFILE ZIP "&folder\tmdb_5000_credits.csv.zip"
member="tmdb_5000_credits.csv";

data myfiles.extracted_credits;
      %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
      infile zipfile delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;
         informat movie_id best32. ;
         informat title $43. ;
         informat cast $28776. ;
         informat crew $22344. ;
         format movie_id best12. ;
         format title $43. ;
        format cast $28776. ;
        format crew $22344. ;
      input
                  movie_id
                  title  $
                  cast  $
                  crew  $
      ;
      if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection
macro variable */
      run;

/*Print first 5 observations for myfiles.extracted.credits */
proc print data=myfiles.extracted_credits (obs=5);
run;
```

b. Merge extracted_movies and extracted_credits

```
/* Merge datasets using PROC SQL */
proc sql;
    create table merged_data as
    select *
    from myfiles.extracted_movies as m
    inner join myfiles.extracted_credits as c
    on m.original_title = c.title
      and m.id = c.movie_id;

quit;
```

c. Create genre dataset

```
/* Create Genres data set*/
data genres;
    set merged_data;

    /* Extract genre names from genres column */
    do i = 1 to countw(genres, '}'); /* Loop through each set of curly braces
in the genres column */
        /* Extract individual genre entry */
        genre_entry = scan(genres, i,'}'); /* Extracting each individual set
of curly braces */
```

```
      do k = 1 to countw(genres, '"');
            genre = scan(genre_entry, 6, '"');
            end;

output;
end;
      keep  title genre;
run;

proc sort data= genres out=genres_sorted;
      by title;
run;

ods rtf bodytitle file ="M:\TermProject\Genres_sorted.rtf";
title "Genres sorted by Movie Title";
proc print data=genres_sorted (obs=10);
run;
ods rtf close;
```

Genres sorted by Movie Title

Obs	title	genre
1	#Horror	Drama
2	#Horror	Mystery
3	#Horror	Horror
4	#Horror	Thriller
5	#Horror	
6	(500) Days of Summer	Comedy
7	(500) Days of Summer	Drama
8	(500) Days of Summer	Romance
9	(500) Days of Summer	
10	10 Cloverfield Lane	Thriller

d. Create Cast dataset

```
/*Create Cast data set*/
data cast;
    set merged_data;

    /* Count the occurrences of "cast_id" in the cast column */
    cast_size = countc(cast, 'cast_id');

    /* Output the observation */
    output;
keep title cast_size
run;

proc sort data= cast out=cast_sorted;
```

```
      by title;
run;

ods rtf bodytitle file ="M:\TermProject\Cast.rtf";
proc print data=cast_sorted (obs=10);
run;
ods rtf close;
```

Cast sorted by Movie Title

Obs	title	cast_size
1	#Horror	304
2	(500) Days of Summer	765
3	10 Cloverfield Lane	322
4	10 Days in a Madhouse	735
5	10 Things I Hate About You	1488
6	102 Dalmatians	151
7	10th & Wolf	400
8	11:14	428
9	12 Angry Men	558
10	12 Rounds	2347

e. Create Budget dataset

```
/*Create Budget data set*/
data budget;
      set merged_data;
      output;
keep title budget;
run;

proc sort data= budget out=budget_sorted;
      by title;
run;

ods rtf bodytitle file ="M:\TermProject\Budget.rtf";
title "Budget sorted by Movie Title";
proc print data=budget_sorted (obs=10);
run;
ods rtf close;
```

Budget sorted by Movie Title

Obs	budget	title
1	1500000	#Horror

Obs	budget	title
2	7500000	(500) Days of Summer
3	15000000	10 Cloverfield Lane
4	1200000	10 Days in a Madhouse
5	16000000	10 Things I Hate About You
6	85000000	102 Dalmatians
7	8000000	10th & Wolf
8	6000000	11:14
9	350000	12 Angry Men
10	20000000	12 Rounds

f. Create Revenue dataset

```
/*Create Revenue data set*/
data revenue;
      set merged_data;
      output;
keep title revenue;
run;

proc sort data= revenue out=revenue_sorted;
      by title;
run;

ods rtf bodytitle file ="M:\TermProject\Revenue.rtf";
title "Revenue sorted by Movie Title";
proc print data=revenue_sorted (obs=10);
run;
ods rtf close;
```

Revenue sorted by Movie Title

Obs	revenue	title
1	0	#Horror
2	60722734	(500) Days of Summer
3	108286421	10 Cloverfield Lane
4	0	10 Days in a Madhouse
5	53478166	10 Things I Hate About You

Obs	revenue	title
6	183611771	102 Dalmatians
7	143451	10th & Wolf
8	0	11:14
9	1000000	12 Angry Men
10	17280326	12 Rounds

g. Merge Genre_sorted, Cast_sorted, Budget_sorted, and Revenue_sorted

```
/*Merge genres_sorted revenue_sorted cast_sorted budget_sorted*/
data combine;
      merge genres_sorted revenue_sorted cast_sorted budget_sorted;
      by title;
run;

proc sort data= combine out=combine_sorted;
      by genre;
run;

data combine_sorted;
    set combine_sorted;
    where not missing(genre);
run;

ods rtf bodytitle file ="M:\TermProject\Combine.rtf";
title "Merge Genre Revenue Budget Cast by Title";
proc print data=combine_sorted (obs=10);
run;
ods rtf close;
```

Merge Genre Revenue Budget Cast by Title

Obs	title	genre	revenue	cast_size	budget
1	10th & Wolf	Action	143451	400	8000000
2	12 Rounds	Action	17280326	2347	20000000
3	13 Hours: The Secret Soldiers of Benghazi	Action	69411370	1142	50000000
4	15 Minutes	Action	56359980	3082	60000000
5	16 Blocks	Action	65664721	687	55000000
6	1941	Action	31755742	1162	35000000
7	2 Fast 2 Furious	Action	236350661	1340	76000000
8	2 Guns	Action	131940411	1493	61000000

Obs	title	genre	revenue	cast_size	budget
9	2012	Action	769653595	1755	200000000
10	21 Jump Street	Action	201585328	638	42000000

h.Write a macro analyze relationship among budget, revenue, cast size by genre

```
/*Write a macro analyze relationship among buget, revenue, cast size
by genre*/
ods rtf file="M:\TermProject\Analyze.rtf";
title "Merge Genre Revenue Budget Cast by Title";
%macro analyze_genre(genre);

    /* Filter data by genre */
       data filtered_data;
            set combine_sorted;
            where genre = "&genre"; /* Use the input parameter genre */
       run;
      /* Calculate statistics */
       title "Summary Statistics for Budget, Revenue, and Cast Size";
       proc means data=filtered_data;
            var budget revenue cast_size;
            output out=summary_stats mean=mean_budget mean=mean_revenue
            mean=mean_cast_size;
       run;

      /* Correlation analysis */
      title "Correlation Matrix for Budget, Revenue, and Cast Size for Genre:
      &genre.";
       proc corr data=filtered_data outp=correlation_matrix;
            var budget revenue cast_size;
       run;

       /* Create scatter plot Budget vs Revenue */
       proc sgplot data=filtered_data;
             title "Relationship between Budget, Revenue for Genre: &genre.";
            scatter x=budget y=revenue / markerattrs=(symbol=circlefilled
            color=blue);
            reg x=budget y=revenue / nomarkers lineattrs=(color=red
            thickness=2);
            xaxis label='Budget ($)';
            yaxis label='Revenue ($)';
       run;

       /* Create scatter plot Revenue vs Cast Size*/
       proc sgplot data=filtered_data;
             title "Relationship between Revenue vs Cast Size for Genre:
            &genre.";
            scatter x=cast_size y=revenue / markerattrs=(symbol=circlefilled
            color=blue);
            reg x=cast_size y=revenue / nomarkers lineattrs=(color=red
            thickness=2);
            xaxis label='Cast Size (people)';
```

```
            yaxis label='Revenue ($)';
        run;

        /* Create scatter plot Budget vs Cast Size */
        proc sgplot data=filtered_data;
              title "Relationship between Budget vs Cast Size for Genre:
             &genre.";
             scatter x=budget y=cast_size / markerattrs=(symbol=circlefilled
             color=blue);
             reg x=budget y=cast_size / nomarkers lineattrs=(color=red
             thickness=2);
             xaxis label='Budget ($)';
              yaxis label='Cast Size (people)';
       run;

        /* Linear Regression Analysis */
       proc reg data=filtered_data ;
       model revenue = budget cast_size;
       title "Linear Regression Analysis for Revenue with Budget and Cast
Size";
       run;

%mend;

%analyze_genre(Action) /* Call the macro with the desired genre */
%analyze_genre(genre)
ods rtf close;
```

Summary Statistics for Budget, Revenue, and Cast Size

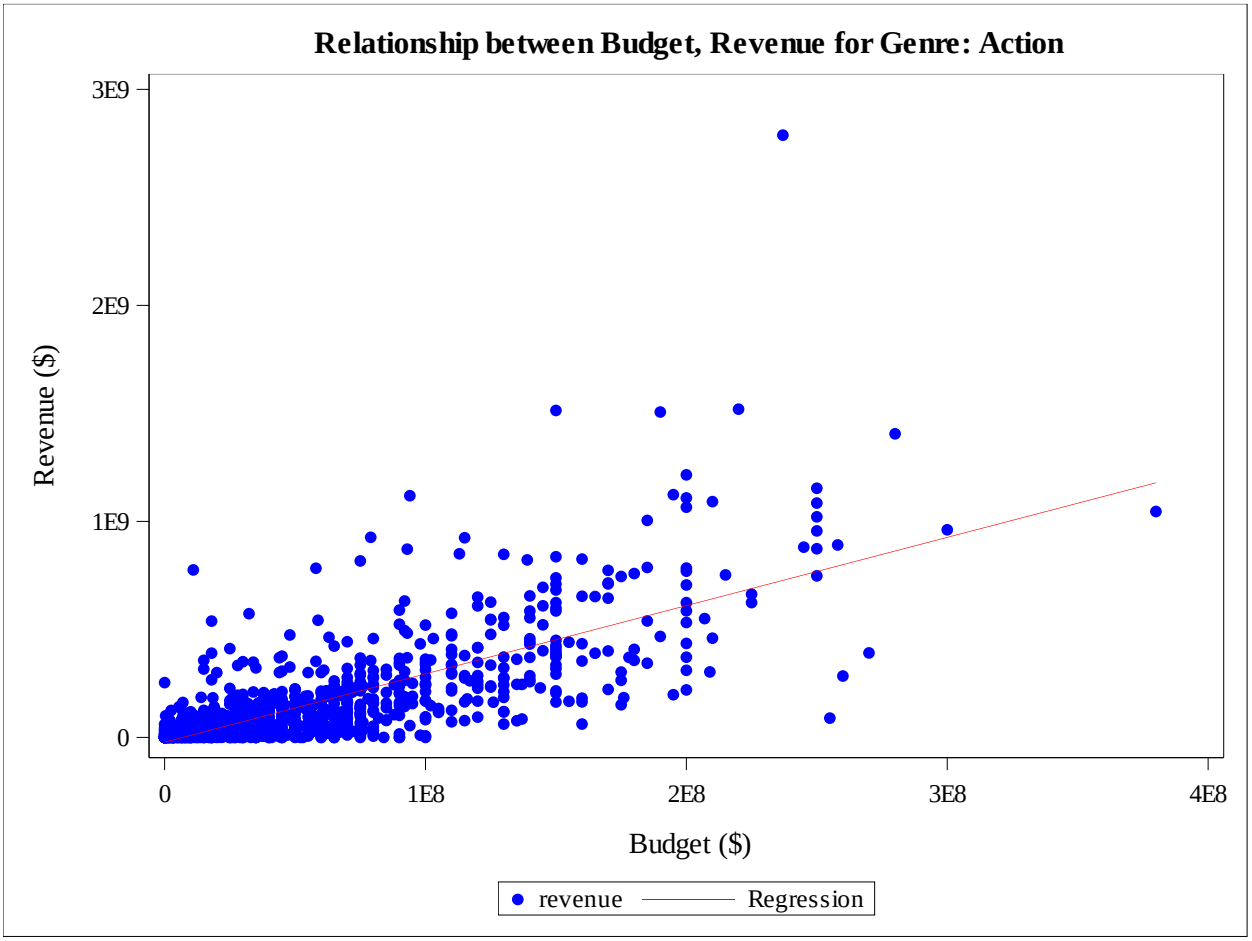
Variable	N	Mean	Std Dev	Minimum	Maximum
budget	1089	53489482.40	55902078.78	0	380000000
revenue	1089	147168795	235544802	0	2787965087
cast_size	1089	912.4811754	796.0115581	0	6575.00

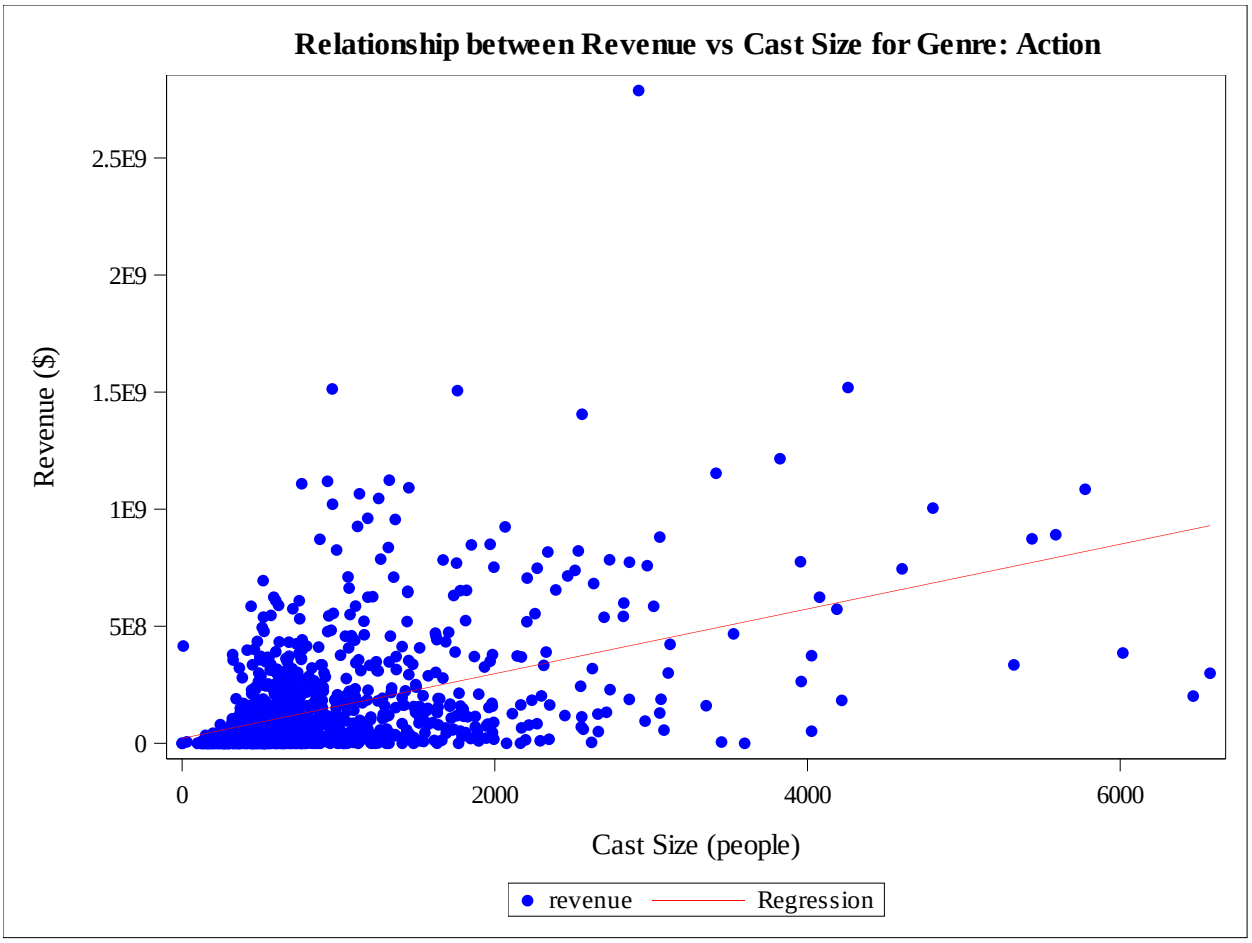
Correlation Matrix for Budget, Revenue, and Cast Size for Genre: Action

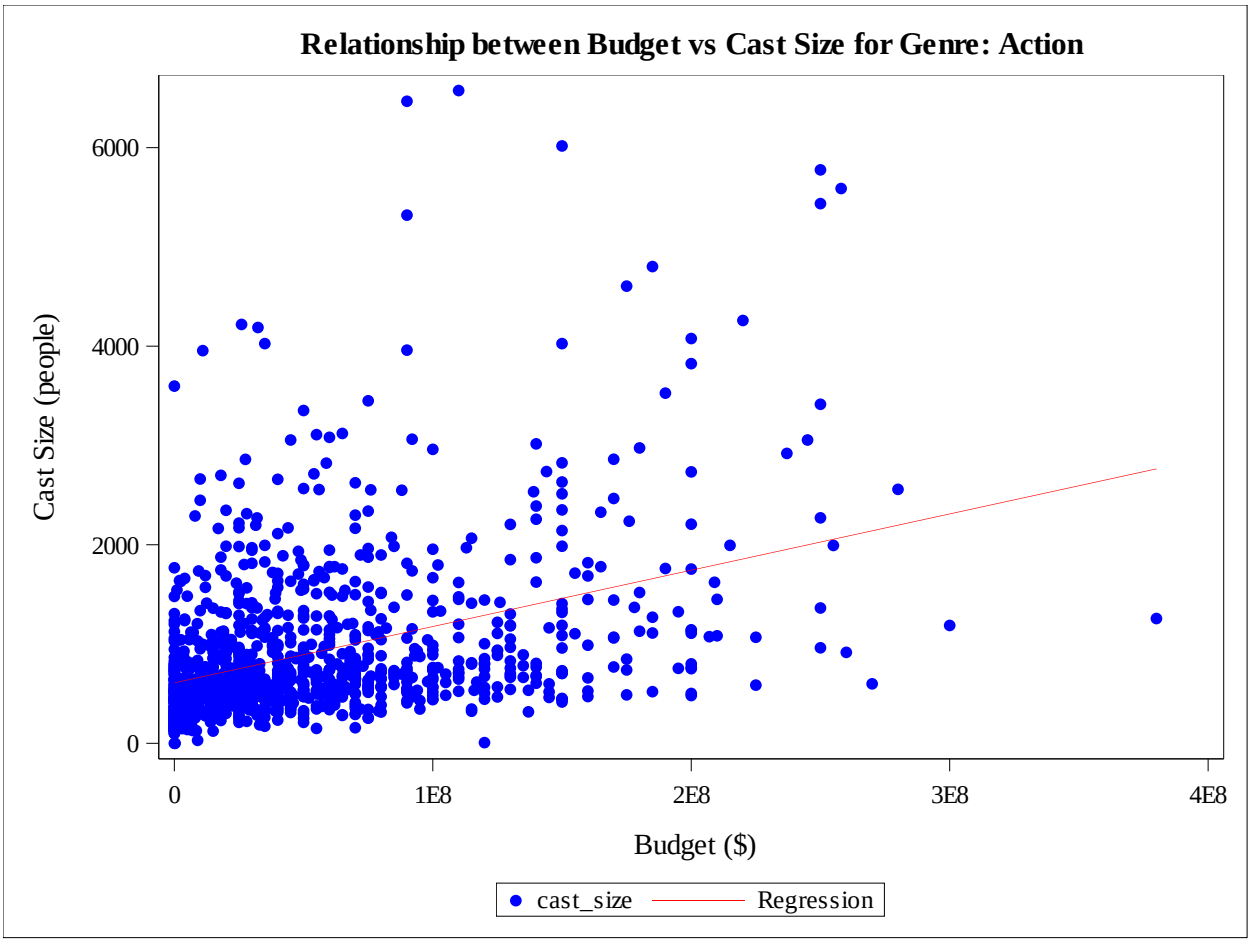
3 Variables:	budget revenue cast_size

Simple Statistics						
Variable	**N**	**Mean**	**Std Dev**	**Sum**	**Minimum**	**Maximum**
budget	1089	53489482	55902079	5.825E10	0	380000000
revenue	1089	147168795	235544802	1.60267E11	0	2787965087
cast_size	1089	912.48118	796.01156	993692	0	6575

Pearson Correlation Coefficients, N = 1089 Prob > |r| under H0: Rho=0			
	budget	**revenue**	**cast_size**
budget	1.00000	0.74990 <.0001	0.39829 <.0001
revenue	0.74990 <.0001	1.00000	0.46702 <.0001
cast_size	0.39829 <.0001	0.46702 <.0001	1.00000




Relationship between Revenue vs Cast Size for Genre: Action
Revenue ($)
2.5E9
2E9
1.5E9
1E9
5E8
0
0
2000
4000
6000
Cast Size (people)
revenue
Regression



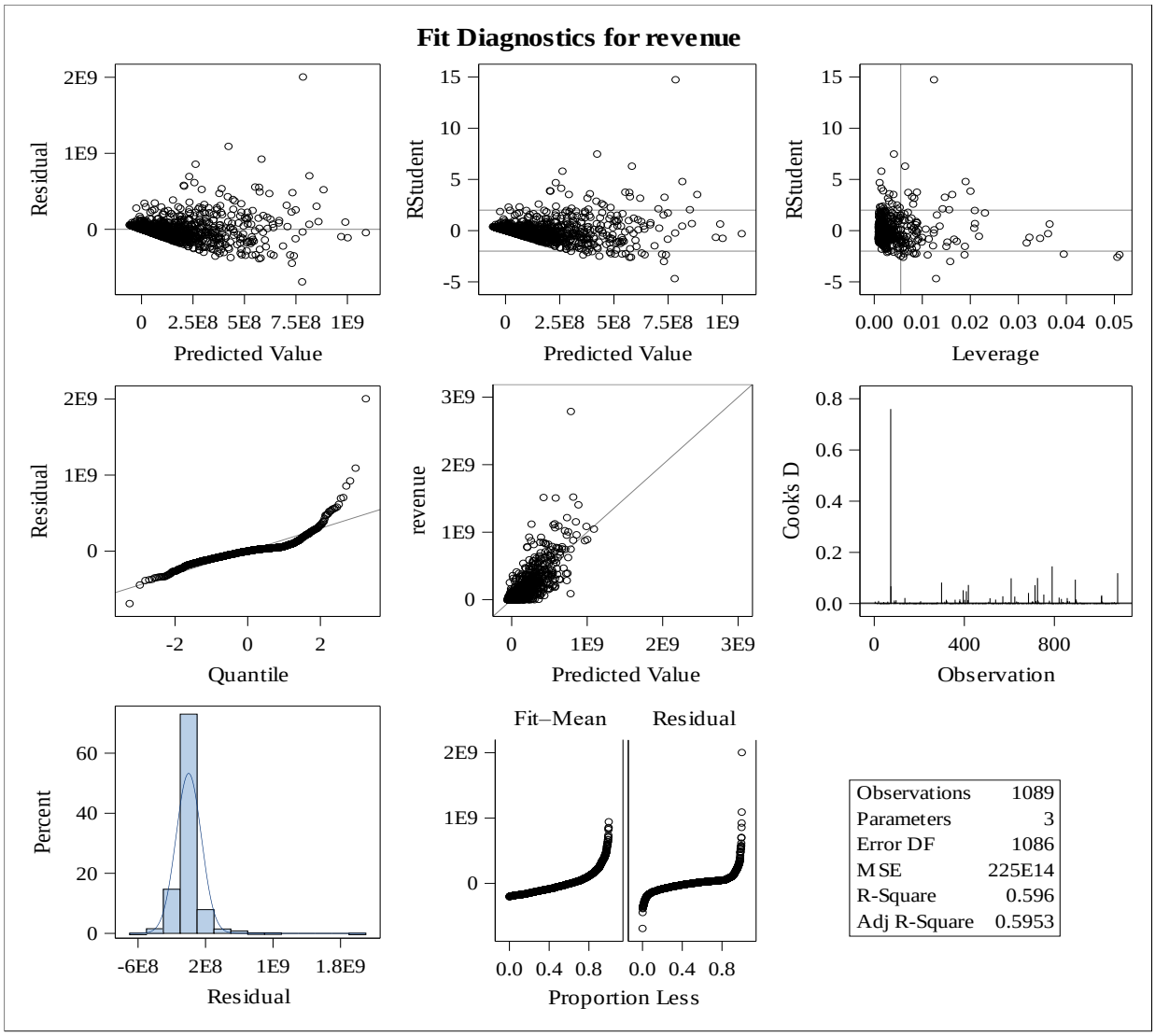
Linear Regression Analysis for Revenue with Budget and Cast Size

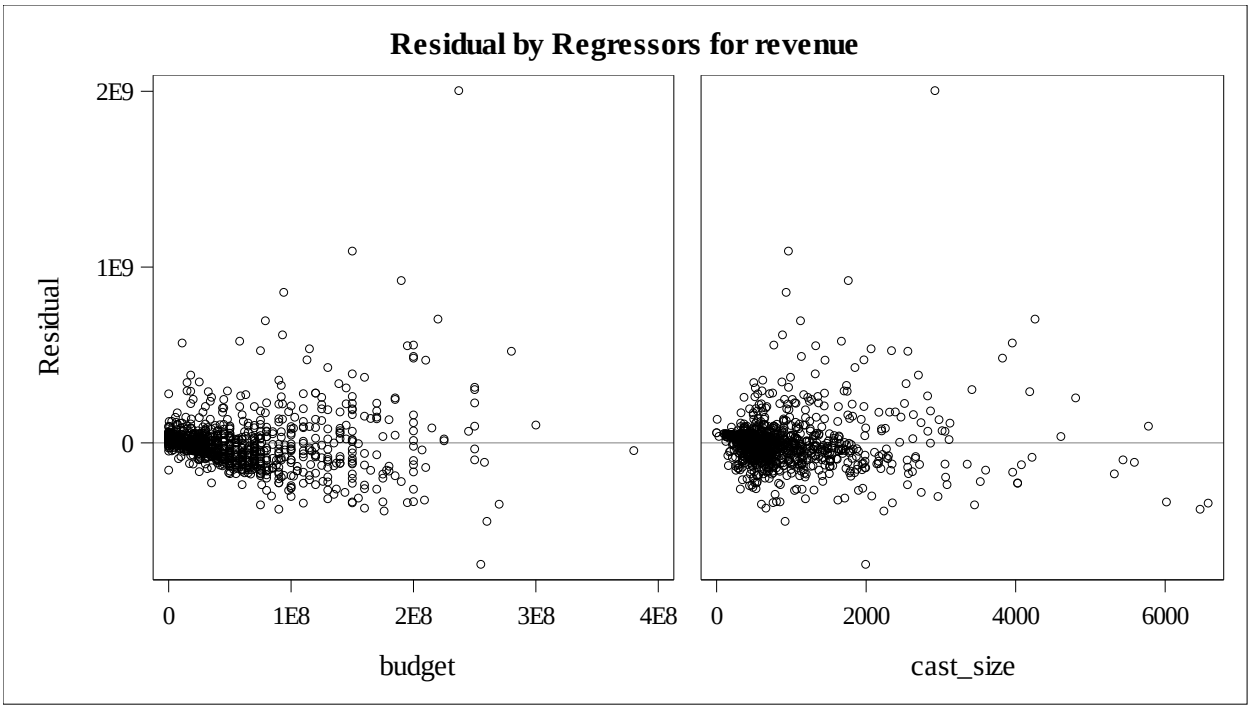
Number of Observations Read	1089
Number of Observations Used	1089

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	2	3.597875E19	1.798937E19	801.17	<.0001
Error	1086	2.438497E19	2.245393E16		
Corrected Total	1088	6.036371E19			

Root MSE	149846348	**R-Square**	0.5960
Dependent Mean	147168795	**Adj R-Sq**	0.5953
Coeff Var	101.81938		

Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	1	-57906151	7340076	-7.89	<.0001
budget	1	2.82395	0.08860	31.87	<.0001
cast_size	1	59205	6221.86718	9.52	<.0001


Fit Diagnostics for revenue
Residual
Predicted Value
RStudent
Predicted Value
RStudent
Leverage
Residual
Quantile
revenue
Predicted Value
Cook's D
Observation
Percent
Residual
Fit–Mean
Residual
Proportion Less
Observations 1089
Parameters 3
Error DF 1086
MSE 225E14
R-Square 0.596
Adj R-Square 0.5953


Residual by Regressors for revenue
Residual
2E9
1E9
0
0
1E8
2E8
3E8
4E8
budget
0
2000
4000
6000
cast_size