D208 Performance Assessment

Alice Tsentsiper

Student ID: 010259639

For this analysis I have chosen to work with Medical Data set. High hospital patient readmission numbers are one of the biggest problems in a medical industry. Hospitals with excessive readmissions are monetary penalized by Centers for Medicare and Medicaid Services, thus the high readmission effects not only patient's health outcomes but hospital's financial stability and reputation. Previous analysis showed a certain correlation between the number of days patient was hospitalized and his higher risk for readmission. In this work I will explore what factors influence longer hospitalization of the patients to try to answer the question what hospital administration can do to mitigate high readmission rates.

Part I: Research Question

A. **Describe the purpose of this data analysis:**

1. **Research question:**

What factors during first hospital stay affect the length of hospitalization upon first admission.

2. **Objectives of the data analysis:**

The goal of the hospital administration should be reduction of cost while preserving the best patient care standards possible. Long hospitalizations cost the hospital in fines, fees and bad reputation, for patients, that may result in poor health outcomes. It is important therefore to determine what factors contribute to patient's prolonged hospital stay and mitigate them.

Part II: Method Justification

B. Multiple regression methods:

1. Multiple regression is a statistical technique used to study the dependance of continues response on two or more linear predictors. If it was determined that the relationship exists, we can then use the model to predict future outcomes given certain independent factors

Multiple regression model assumptions:

1. Linear relationship exists between the dependent variable and the independent variables, linear relationship can be tested by scatterplots through trend line to see dependency.
2. Multiple regression assumes normal distribution of residuals, tested by plotting a histogram of the residuals from regression model (ref 2).
3. Multicollinearity: Independent variables are not correlated with each other (tested with Variance Inflation Factor) (ref 1).
4. It requires at least two independent variables, nominal, ordinal or interval.
5. Homoscedasticity- the distribution of errors in the independent variables is similar. It could be checked by plotting residuals (the difference between predicted and actual values=errors)

2. Benefits of using Python in the data analysis:

The user-friendly syntax and preprogrammed extensive libraries for data analysis were the reason for my choice to use Python. Python is general-purpose language that is widespread, has a great readability and clear syntax. Python is a language that can be used and integrated with multiple platforms therefore eliminating the need to transform or manipulate the data after the analysis is done. Python has extensive libraries designed specifically for data analysis that can be used in every stage of statistical analysis. The main libraries used for data processing and mining are NumPy and Pandas that deal with large sets of data., Sklearn library provides advanced analytics tools for machine learning and Matplotlib and Seaborn libraries are used to create easy to understand graphics.

3. Multiple linear regression as appropriate tool for this analysis:

Multiple linear regression is used to estimate the relationship between two or more independent variables, and one continues quantitative dependent variable. Since our dependent variable is days of initial hospitalization presented as continues data type, we can use multiple linear regression when trying to answer:

- How strong the relationship is between two or more independent variables and one dependent variable for example how age, preexisting health conditions and complication risk affect hospitalization days.
- The value of the dependent variable at a certain value of the independent variables (e.g. the expected days of hospitalization at certain age, health conditions and complication risks).

Upon establishing whether linear relationship exists by multiple linear regression, we can then try to identify the patients with a risk for prolonged hospital stay through predictive modeling.

Part III: Data Preparation

C. **Summarize the data preparation process:**

1. Data preparation goals and techniques:

Data preparation goals is to clean and transform data in order to perform analytics and run a prediction model on an accurate and meaningful data.

The data preparation process will include:

- Loading libraries
- Reading the data file into pandas
- Examine data structure and shape, examine column names
- Detection and elimination of duplicates,
- Data types -overview, transform data from categorical to numeric with get dummies method to use in a model.
- Elimination of any inconsistencies in format, spelling data type representation:
- Changing column names to more descriptive strings, make sure all titles start with capitals, dates are in similar format, prices represented as integers and etc.,
- Detection and treatment: fill in any missing values with zero, mean or median, identify and treat outliers by replacing extreme values with median.

2. **Summary statistics**:

The medical data set is used to determine what factors are contribute to prolonged hospitalization. For this analysis I have not taken in to the account geographic variables such as address of the patient. I have also did not include the gender, marital status, income or the charges for the hospital stay as I felt they would not add any contribution to the analysis. We have 10000 samples ,31 predictive variables and 1 variable that is dependent for this question.

The target variable is the number of days a patient was in the hospital upon initial hospitalization. Column name is Initial_days and the data presented as continues data type.

My hypothesis is that there are factors that influence how much time the patient is spending in the hospital. Those independent factors may include patients' physical characteristics like the patient's age, severity of patients' condition upon admission such as complication risk, services that were provided to him, the way he was initially admitted, his overall health conditions (ex-Overweight, Diabetes etc) presented as binary data of yes or no and the description of his stay as described in the survey Item 1 to 8, thus the independent variables I have chosen to focus on are;

Variable	Data Type	Description	Examples
Children	quantitive/o	how many children patient has	3

	rdinal		
Age	continues	age of the patient	53
ReAdmis	categorica/qualitive	Whether the patient was readmitted within a month of release or not	yes/no
VitD_levels	continues	levels mesured in ng/ml	17.80233
Doc_visits	Continues/ quantitive	Number of times the primary physician visited the patient during the initial hospitalization	6
Full meals eaten	discriptive	Number of full meals the patient ate while hospitalized	0 to 7
VitD_supp	ordinal/continues	The number of times that vitamin D supplements were administered to the patient	0 to 5
Soft Drink	categorical	does the patient drinks three or more sodas a day	yes/no
Initial_admn	categorical/ nominal	The means by which the patient was admitted into the hospital initially	Elective/Observation/Emergency
High Blood	Categorical nominal	does the patient has a high blood pressure	yes/no
Stroke	Categorical nominal	has the patient had a stroke	yes/no
Complication_risk	Categorical nominal	Level of complication risk for the patient as assessed by a primary patient assessment	high/ medium/low
Overweight	Categorical nominal	is the patient overweight	0/1
Arthritis	Categorical nominal	does the patient has the following condition	yes/no
Diabetes	Categorical nominal	does the patient has the following condition	yes/no
Hyperlipidemia	Categorical nominal	does the patient has the following condition	yes/no
BackPain	Categorical nominal	does the patient has the following condition	yes/no
Anxiety	Categorical nominal	does the patient has the following condition	0/1
Allergic_rhinitis	Categorical nominal	does the patient has the following condition	yes/no
Reflux_esophagitis	Categorical nominal	does the patient has the following condition	yes/no
Asthma	Categorical nominal	does the patient has the following condition	yes/no
Services	ordinal/discrete	Primary service the patient received while hospitalized	CT SCAN/ BLOOD WORK/ Intravenous/ MRI
Item1	ordinal/discrete	Customer rating of timely admission	scale of 1 to 8
Item2	ordinal/discrete	Customer rating of timely treatment	scale of 1 to 8
Item3	ordinaldiscrete	Customer rating of timely visits	scale of 1 to 8
Item4	ordinaldiscrete	Customer rating of Reliability	scale of 1 to 8
Item5	ordinal/discrete	Customer rating of options	scale of 1 to 8
Item6	ordinal/discrete	Customer rating of hours of treatment	scale of 1 to 8
Item7	ordinal/discrete	Customer rating of Courteous staff	scale of 1 to 8

Item8	ordinal/discrete	evidence of active listening of doctor	scale of 1 to 8

Quick look at the descriptive statistics can show us any abnormalities in the selected information. We can see standard deviation in the data, the min and max values and the mean of the data. When examining the Initial_days column, we see that it has a nice distribution from 1 to 72 with a mean of 34.46, and 50% at 35.84 days which is very close to the mean.

Descriptive statistics:

Column1	Children	Age	VitD_levels	Doc_visits
Count	10000.00	10000.00	10000.00	10000.00
mean	2.10	53.51	17.96	5.01
std	2.16	20.64	2.02	1.05
min	0	18	9.806483	1
25%	0	36	16.626439	4
50%	1	53	17.951122	5
75%	3	71	19.347963	6
max	10	89	26.394449	9

Column1	Full_meals_eaten	vitD_supp	Initial_admin	Complication_risk	Services	Initial_days
Count	10000.00	10000.00	10000.00	10000.00	10000.00	10000.00
mean	1.00	0.40	2.26	2.12	1.67	34.46
std	1.01	0.63	0.83	0.73	0.83	26.31
min	0	0	1	1	1	1.00
25%	0	0	2	2	1	7.90
50%	1	0	3	2	1	35.84
75%	2	1	3	3	2	61.16
max	7	5	3	3	4	71.98

Column1	Timely_Admission	Timely_Treatment	Timely_visits
Count	10000.00	10000.00	10000.00
mean	3.52	3.51	3.51
std	1.03	1.03	1.03
min	1	1	1
25%	3	3	3
50%	4	3	4
75%	4	4	4

max	8	7	8

	Reliability	Options	Hours_of_Treatment	Courteous_Staff	Active_Listening	ReAdmis	Soft_drink	HighBlod	Stroke
count	10000	10000	10000	10000	10000	10000	10000	10000	10000
mean	3.5151	3.4969	3.5225	3.494	3.5097	0.3669	0.2575	0.409	0.1993
std	1.036282	1.030192	1.032376	1.021405	1.042312	0.481983	0.437279	0.491674	0.399494
min	1	1	1	1	1	0	0	0	0
25%	3	3	3	3	3	0	0	0	0
50%	4	3	4	3	3	0	0	0	0
75%	4	4	4	4	4	1	1	1	0
max	7	7	7	7	7	1	1	1	1

	Overweight	Arthritis	Diabetes	Hyperlipidemia	BackPain
count	10000	10000	10000	10000	10000
mean	0.7094	0.3574	0.2738	0.3372	0.4114
std	0.454062	0.479258	0.44593	0.472777	0.492112
min	0	0	0	0	0
25%	0	0	0	0	0
50%	1	0	0	0	0
75%	1	1	1	1	1
max	1	1	1	1	1

	Anxiety	Allergic_rhinitis	Reflux_esophagitis	Asthma
count	10000	10000	10000	10000
mean	0.3215	0.3941	0.4135	0.2893
std	0.467076	0.488681	0.492486	0.45346
min	0	0	0	0
25%	0	0	0	0
50%	0	0	0	0
75%	1	1	1	1
max	1	1	1	1

3. **Explain and show Annotated code**:

```
# Export libraries and read dataset into Pandas:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
data=pd.read_csv('medical_clean.csv', index_col='CaseOrder')
```

~~~~~~

```
# Examine the data, look for the data structure, size, column names:
data.head()
data.info()
data.shape()
data.columns
# Look for duplicates:
data.duplicated().sum()
#Looking for missing values:
data.isnull().sum()
```

Summary: There are no duplicates in the data, there are no missing values. It appears the data was cleaned.

~~~~~~

```
# Rename columns: Item 1 to Item 8 to more descriptive names:
new_col_names={'Item1':'Timely_Admission', 'Item2':'Timely_Treatment',
'Item3':'Timely_visits', 'Item4':'Reliability', 'Item5':'Options',
'Item6':'Hours_of_Treatment', 'Item7':'Courteous_Staff', 'Item8':'Active_Listening'}
data.rename(columns=new_col_names, inplace=True)
data.columns
```

```
Index(['Customer_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip',
       'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Children',
       'Age', 'Income', 'Marital', 'Gender', 'ReAdmis', 'VitD_levels',
       'Doc_visits', 'Full_meals_eaten', 'vitD_supp', 'Soft_drink',
       'Initial_admin', 'HighBlood', 'Stroke', 'Complication_risk',
       'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain',
       'Anxiety', 'Allergic_rhinitis', 'Reflux_esophagitis', 'Asthma',
       'Services', 'Initial_days', 'TotalCharge', 'Additional_charges',
       'Timely_Admission', 'Timely_Treatment', 'Timely_visits', 'Reliability',
       'Options', 'Hours_of_Treatment', 'Courteous_Staff', 'Active_Listening'],
      dtype='object')
```

~~~~~~

 # Removing less meaningful columns from the data:

data=data.drop(columns=['Customer_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Marital', 'TotalCharge', 'Additional_charges', 'Income', Gender])

 # Check the new data shape

data.shape

(1000,32)

~~~~~~

#Changing object dtype to integer with dummy values

```
binary_columns=['ReAdmis','Soft_drink', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis', 'Reflux_esophagitis', 'Asthma']

for col in binary_columns:
  data_d=pd.get_dummies(data[col], prefix=col, columns=col, drop_first=True)
  data = pd.merge(
    left=data,
    right=data_d,
    left_index=True,
    right_index=True,
  )
  data = data.drop(columns=col)
print(data.columns)
```

```
Index(['Children', 'Age', 'Gender', 'VitD_levels', 'Doc_visits',
       'Full_meals_eaten', 'vitD_supp', 'Initial_admin', 'Complication_risk',
       'Services', 'Initial_days', 'Timely_Admission', 'Timely_Treatment',
       'Timely_visits', 'Reliability', 'Options', 'Hours_of_Treatment',
       'Courteous_Staff', 'Active_Listening', 'ReAdmis_Yes', 'Soft_drink_Yes',
       'HighBlood_Yes', 'Stroke_Yes', 'Overweight_Yes', 'Arthritis_Yes',
       'Diabetes_Yes', 'Hyperlipidemia_Yes', 'BackPain_Yes', 'Anxiety_Yes',
       'Allergic_rhinitis_Yes', 'Reflux_esophagitis_Yes', 'Asthma_Yes'],
      dtype='object')
```

 # Rename the dummy columns with original names:

```
data.rename(columns = {'ReAdmis_Yes':'ReAdmis', 'Soft_drink_Yes':'Soft_drink',
'HighBlood_Yes':'HighBlod', 'Stroke_Yes':'Stroke', 'Overweight_Yes':'Overweight',
'Arthritis_Yes':'Arthritis',

'Diabetes_Yes':'Diabetes', 'Hyperlipidemia_Yes':'Hyperlipidemia',
'BackPain_Yes':'BackPain', 'Anxiety_Yes':'Anxiety',

'Allergic_rhinitis_Yes':'Allergic_rhinitis', 'Reflux_esophagitis_Yes':'Reflux_esophagitis',
'Asthma_Yes':'Asthma' }, inplace = True)

data.columns
```

```
Index(['Children', 'Age', 'Gender', 'VitD_levels', 'Doc_visits',
       'Full_meals_eaten', 'vitD_supp', 'Initial_admin', 'Complication_risk',
       'Services', 'Initial_days', 'Timely_Admission', 'Timely_Treatment',
       'Timely_visits', 'Reliability', 'Options', 'Hours_of_Treatment',
       'Courteous_Staff', 'Active_Listening', 'ReAdmis', 'Soft_drink',
       'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes',
       'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic_rhinitis',
       'Reflux_esophagitis', 'Asthma'],
      dtype='object')
```

Replace method to replace strings to numeric in Service, Complication_Risk and Initial_admin, Gender:

```
data['Services'].replace(['Blood Work', 'Intravenous', 'CT Scan', 'MRI' ], [1,2,3,4], inplace=True)

data['Initial_admin'].replace(['Emergency Admission', 'Elective Admission', 'Observation
Admission'], [3,2,1], inplace=True)

data['Complication_risk'].replace(['Low', 'Medium', 'High'], [1,2,3], inplace =True)

data['Gender'].replace (['Male', 'Female', 'Prefer not to answer'], [1,2,0], inplace=True)
```

 # Get descriptive statistics, examine the data for outliers:

```
data.describe()
```

```
# Visualize distributions through boxplot or histogram and decide on outlier treatment; fill in with median, quartile value or do not change.

sns.boxplot(x='Initial_admin', y='Initial_days',data=data)

sns.distplot(data['Initial_days'])

plt.show()
```

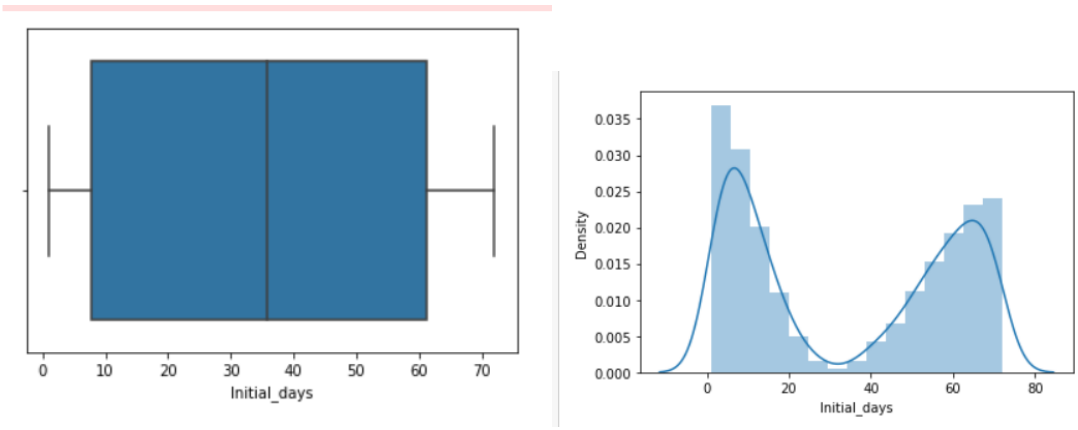
```
#Save the cleaned data in csv file.

data.to_csv('medical_data_clean.csv')
```

4. Univariate and Bivariate visualizations:

Univariant analysis:

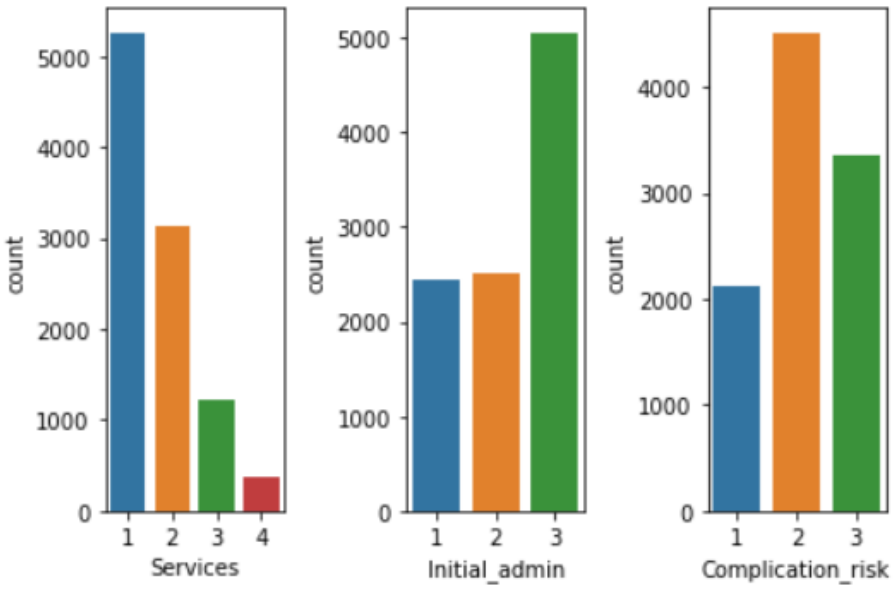
Our response variable of interest Initial_days (how many days the patient spend in the hospital) has no outliers. There are no outliers however the data is not normally distributed according to the histogram.

```
sns.boxplot('Initial_days', data=data)

plt.show()

sns.distplot(data['Initial_days'])
```



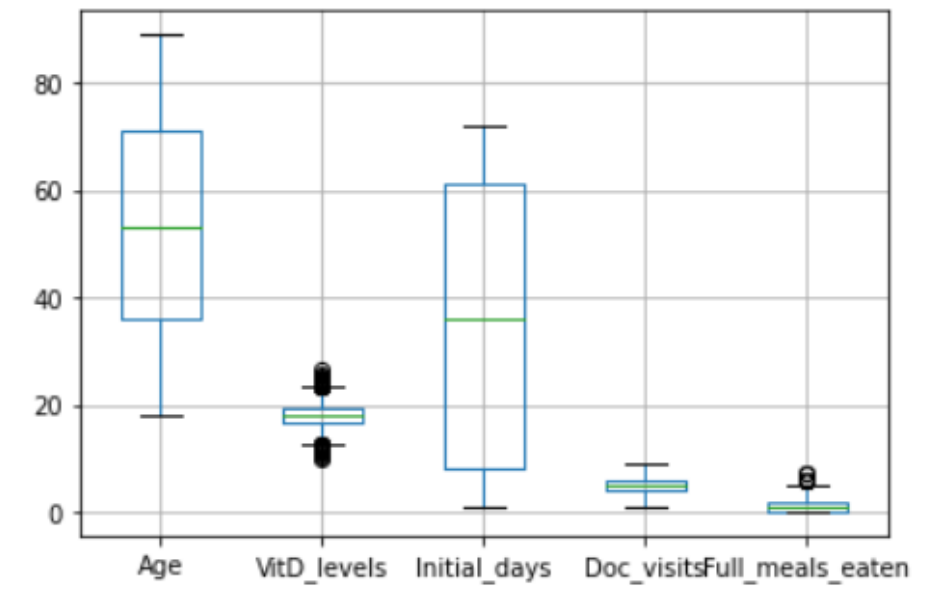

```
fig, ax=plt.subplots(1,3)
sns.countplot('Services', data=data, ax=ax[0])
sns.countplot('Initial_admin', data=data, ax=ax[1])
sns.countplot('Complication_risk', data=data, ax=ax[2])
plt.show()
```

Services, Initial_admin and Complication_risk are categorical values transformed to numeric.



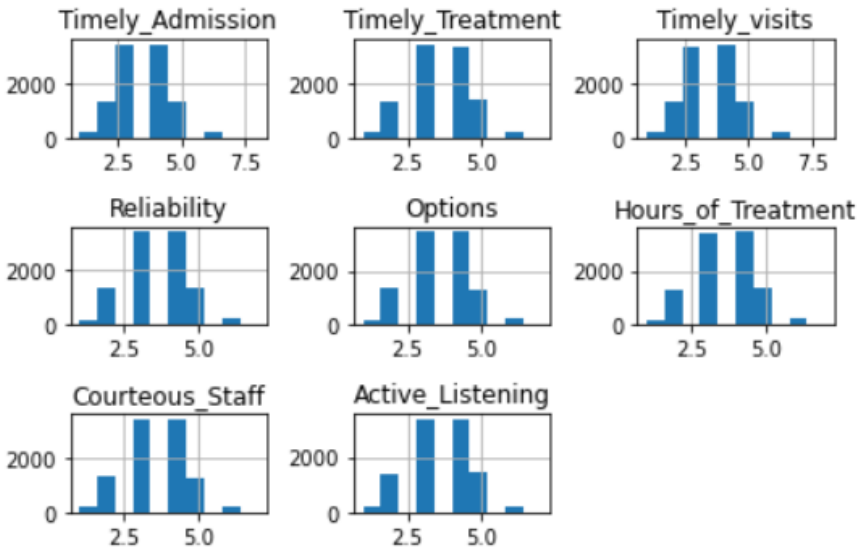
Distribution of continues variables in the boxplot form

```
data[['Age', 'VitD_levels', 'Initial_days', 'Doc_visits', 'Full_meals_eaten' ]].hist()
plt.show()
```

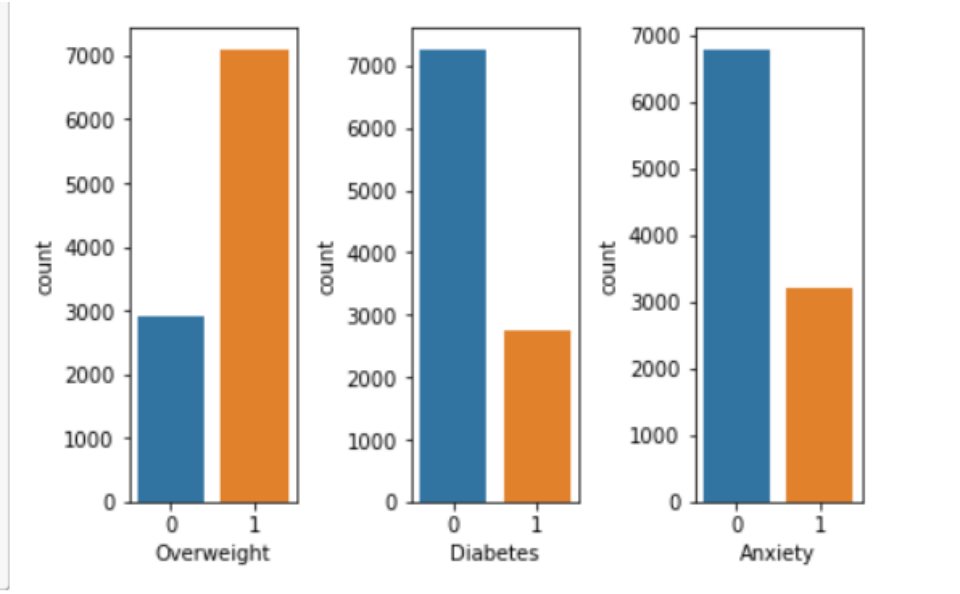



```
data[['Timely_Admission', 'Timely_Treatment',
      'Timely_visits', 'Reliability', 'Options', 'Hours_of_Treatment',
      'Courteous_Staff', 'Active_Listening']].hist()
plt.show()
```

Distribution of the survey responses appears to be normal.




```
fig, ax =plt.subplots(1,3)
sns.countplot(data['Overweight'], ax=ax[0])
sns.countplot(data['Diabetes'], ax=ax[1])
sns.countplot(data['Anxiety'], ax=ax[2])
Plt.tight_layout()
fig.show()
```

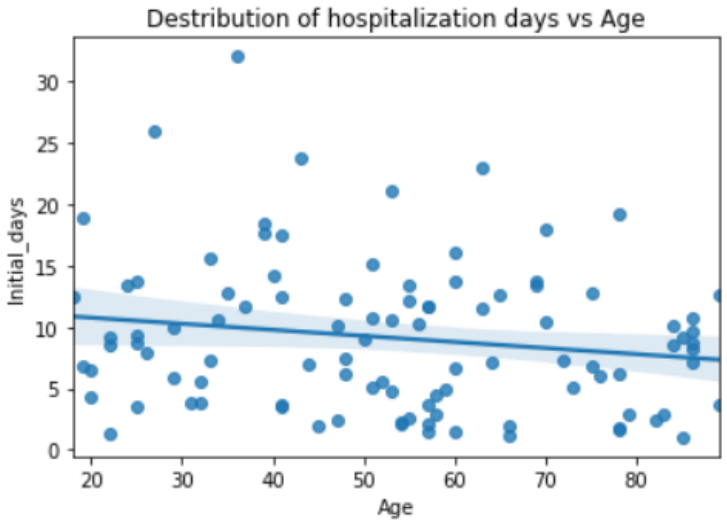


Bivariant analysis

#Bivarient analysis;

#Reduced sample size to see distribution of hospitalization days with Age

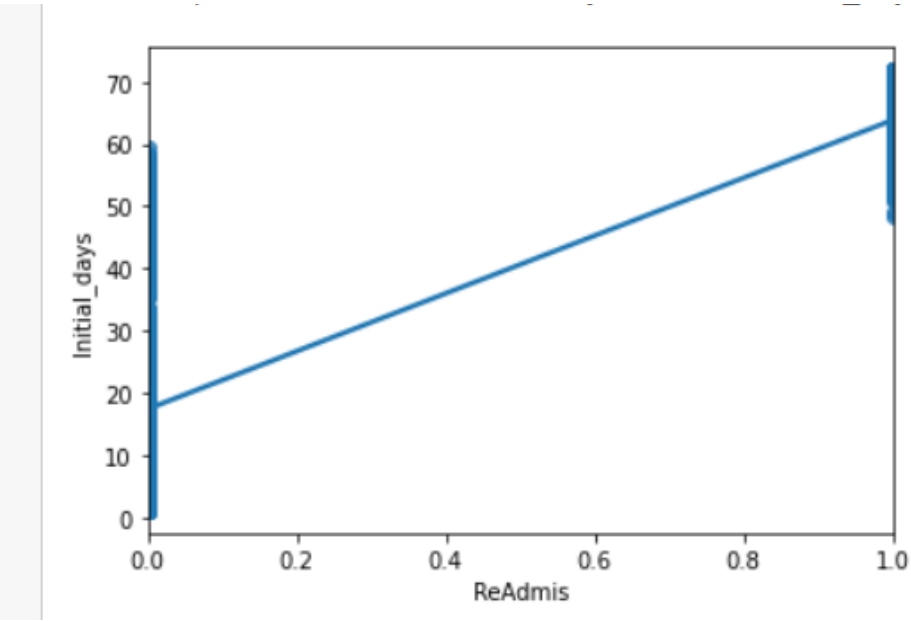
```
data_sample=data.head(100)
sns.regplot(x="Age", y="Initial_days", data=data_sample)
plt.title('Destribution of hospitalization days vs Age')
plt.show()
```



There does not seem to be strong correlation between age and initial hospitalization

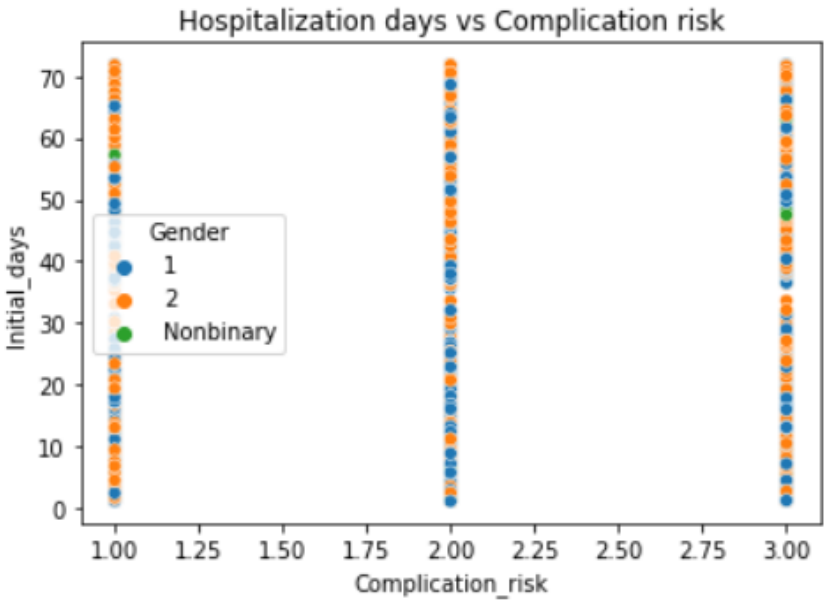
#Corelation between Readmission chances and Initial days in the hospital

```
sns.regplot(x='ReAdmis', y='Initial_days', data=data)
```



There is strong coliniarity between Initial days and ReAdmission risk .

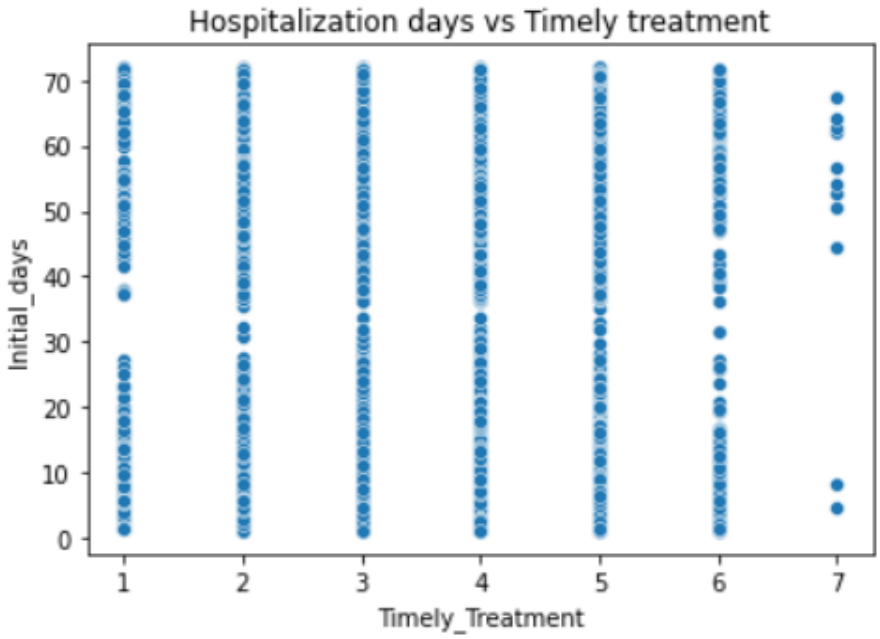
```
sns.scatterplot(x='Complication_risk', y='Initial_days',hue='Gender', data=data)
plt.title('Hospitalization days vs Complication risk')
plt.show()
```



Gender= 1-male, 2- Female

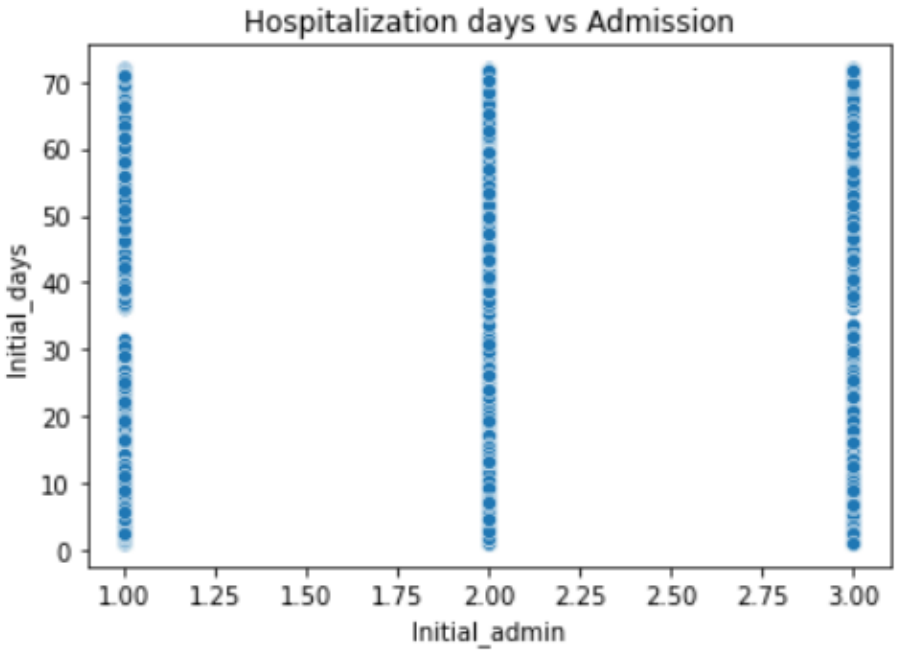
There might not be correlation between complication risk and initial days at the hospital

```
sns.scatterplot(x='Timely_Treatment', y='Initial_days', data=data)
plt.title('Hospitalization days vs Timely treatment')
plt.show()
```



Timely treatment is distributed equally across hospitalization days except score 7 that seems to be more associated with the high number of days.

```
sns.scatterplot(x='Initial_admin', y='Initial_days',data=data)
plt.title('Hospitalization days vs Admission' )
plt.show()
```

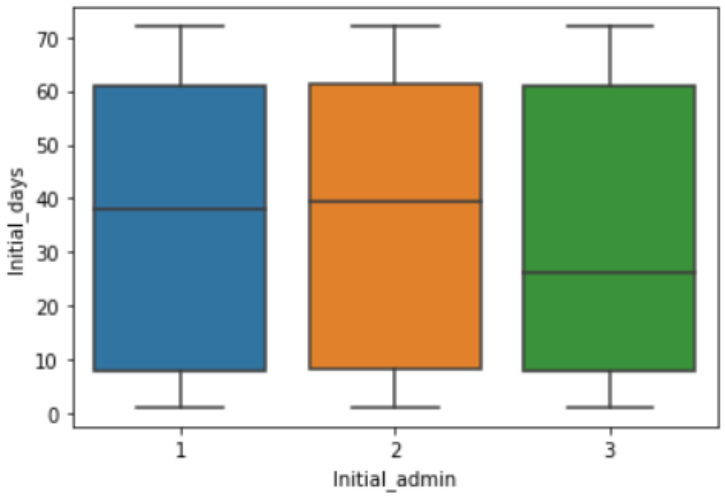


#Emergency Admission'=3, 'Elective Admission'=2, 'Observation Admission'=1

sns.boxplot(x='Initial_admin', y='Initial_days',data=data)

plt.show()

#Emergency Admission'=3, 'Elective Admission'=2, 'Observation Admission'=1



Emergency admission has a lower median then the rest of the admissions, perhaps has significance.

5. Provide a copy of the prepared data set.

Copy of the cleaned data is attached in the task as a link.

Part IV: Model Comparison and Analysis

For this model I have omitted the ReAdmis variable. It indicates whether the patient was readmitted to the hospital after the first admission. It cannot have effect on the initial days of hospitalization but bivariant analysis shows that the number of days in the hospital may affect the risk for repeat admission.

D. 1. Initial multiple regression model from *all* predictors that were identified in Part C2.

#Instal packages:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

from statsmodels.formula.api import ols

clean_data=pd.read_csv('medical_data_clean.csv')

clean_data=clean_data.drop(columns=['CaseOrder'])

clean_data.columns

clean_data.shape

Initial Multiple Linear Regression Model:

 # Runing initial model that includes all the columns

model_data=ols('Initial_days~ Children + Age+ VitD_levels+ Doc_visits +Full_meals_eaten +vitD_supp+ Initial_admin+Complication_risk+
Services+Timely_Admission+Timely_Treatment+Timely_visits+Reliability +

Options+Hours_of_Treatment+Courteous_Staff+Active_Listening+Soft_drink +HighBlood+Stroke+ Overweight+ Arthritis+Diabetes+Hyperlipidemia+ BackPain+ Anxiety+ Allergic_rhinitis+ Reflux_esophagitis+Asthma', data=clean_data).fit()

print(model_data.summary())

The model statistics:

OLS Regression Results

Dep. Variable:	Initial_days	R-squared:	0.004
Model:	OLS	Adj. R-squared:	0.001
Method:	Least Squares	F-statistic:	1.317
Date:	Tue, 05 Jul 2022	Prob (F-statistic):	0.119
Time:	12:44:38	Log-Likelihood:	-46869.
No. Observations:	10000	AIC:	9.380e+04
Df Residuals:	9970	BIC:	9.401e+04
Df Model:	29		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	37.7926	3.845	9.829	0.000	30.256	45.329
Children	0.2716	0.122	2.232	0.026	0.033	0.510
Age	0.0203	0.013	1.593	0.111	-0.005	0.045
VitD_levels	-0.0315	0.131	-0.241	0.810	-0.288	0.225
Doc_visits	-0.1798	0.252	-0.714	0.475	-0.673	0.314
Full_meals_eaten	-0.4421	0.261	-1.692	0.091	-0.954	0.070
vitD_supp	0.6671	0.419	1.592	0.111	-0.154	1.488
Initial_admin	-0.2902	0.319	-0.909	0.364	-0.916	0.336
Complication_risk	-0.4770	0.361	-1.323	0.186	-1.184	0.230
Services	0.0673	0.316	0.213	0.832	-0.553	0.687
Timely_Admission	-0.8244	0.379	-2.174	0.030	-1.568	-0.081
Timely_Treatment	0.2611	0.350	0.746	0.455	-0.425	0.947
Timely_visits	0.0347	0.323	0.107	0.914	-0.598	0.667
Reliability	-0.3598	0.288	-1.251	0.211	-0.923	0.204
Options	0.0161	0.303	0.053	0.958	-0.578	0.610
Hours_of_Treatment	-0.0538	0.313	-0.172	0.863	-0.667	0.559
Courteous_Staff	0.3519	0.294	1.195	0.232	-0.225	0.929
Active_Listening	-0.0575	0.277	-0.208	0.836	-0.601	0.486
Soft_drink	0.1644	0.603	0.273	0.785	-1.017	1.346
HighBlood	-0.3438	0.536	-0.642	0.521	-1.394	0.706
Stroke	-0.1264	0.659	-0.192	0.848	-1.418	1.165
Overweight	-0.5901	0.580	-1.018	0.309	-1.727	0.547
Arthritis	1.0286	0.550	1.872	0.061	-0.049	2.106
Diabetes	-0.1239	0.591	-0.210	0.834	-1.283	1.035
Hyperlipidemia	-0.1959	0.557	-0.352	0.725	-1.288	0.897
BackPain	0.9117	0.535	1.703	0.089	-0.138	1.961
Anxiety	0.6454	0.564	1.145	0.252	-0.460	1.750
Allergic_rhinitis	0.2157	0.539	0.400	0.689	-0.841	1.272
Reflux_esophagitis	0.6709	0.535	1.254	0.210	-0.378	1.719
Asthma	-0.7957	0.581	-1.370	0.171	-1.934	0.343

Omnibus:	41652.924	Durbin-Watson:	0.166
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1270.719
Skew:	0.070	Prob(JB):	1.17e-276
Kurtosis:	1.259	Cond. No.	892.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Regression Formula:

y = 22.69 + 0.027 * Children + 0.002* Age + -0.03*VitD_levels -0.17*Doc_visits -0.4*Full_meals_eaten + 0.6*vitD_supp -0.39*Initial_admin -0.47*Complication_risk +0.06*Services -0.8*Timely_Admission+0.26Timely_Treatment + 0.03Timely_visits -0.35*Reliability +0.06Options -0.05Hours_of_Treatment + 0.3Courteous_Staff - 0.05 Active_Listening -0.34HighBlood+0.16Soft_drink -0.126Stroke -0.59 Overweight+ 1.02Arthritis -0.12Diabetes-0.20Hyperlipidemia +0.9BackPain + 0.64Anxiety + 0.21Allergic_rhinitis +0.67Reflux_esophagitis-0.79Asthma

Validating the initial model:

 # Looking at the coefficient determination. It is the factor that explaining the portion of variations in the dependent variable that comes from the independent variable. R^2 score is ranging between 0 and 1, the closer it is to one the greater is the linear relationship between the variables.

print('R2 score:', model_data.rsquared)

```
R2 score: 0.003815829743797683
```

R^2 score of 0.038 means 0. 4 % of the variation in our dependent variable can be explained using our independent variables. Looking at the adjusted R square we can see it is even lower at 0.1%. This is very low R number, and it is likely that none of the independent variables influence the number of hospitalization days in a linear relationship.

 # F-test checks independent variables combined as related to the dependent variable when compared to the state when all the independent variables are 0. If F-statistic p value is greater than 0.05, there is no evidence of co linearity between combined independent variable with the output

print('F-statistic:', model_data.fvalue)

print('Probability of observing value at least as high as F-statistic:', model_data.f_pvalue)

```
F-statistic: 1.3168809433230266
Probability of observing value at least as high as F-statistic:
0.11864182587747099
```

Conclusion: F p value higher than alpha 0.05 means that our independent variables combine have no correlation to the dependent variable

 # Looking at p-Values : independent variables with p values lower than 0.05 will be the most influential on the target variable. Observations with higher p_values can be taken out for the reduced model.

print('p- values :', model_data.pvalues)

```
p- values : Intercept            1.066882e-22
Children                2.565722e-02
Age                     1.112780e-01
VitD_levels             8.095092e-01
Doc_visits              4.752569e-01
Full_meals_eaten        9.073976e-02
vitD_supp               1.113649e-01
Initial_admin           3.636255e-01
Complication_risk       1.860026e-01
Services                8.315262e-01
Timely_Admission        2.969167e-02
Timely_Treatment        4.554369e-01
Timely_visits           9.144763e-01
Reliability             2.108317e-01
Options                 9.575797e-01
Hours_of_Treatment      8.633913e-01
Courteous_Staff         2.320992e-01
Active_Listening        8.356199e-01
Soft_drink              7.849952e-01
HighBlood               5.210159e-01
Stroke                  8.478636e-01
Overweight              3.089070e-01
Arthritis               6.127904e-02
Diabetes                8.340309e-01
Hyperlipidemia          7.252017e-01
BackPain                8.862610e-02
Anxiety                 2.522655e-01
Allergic_rhinitis       6.889268e-01
Reflux_esophagitis      2.097387e-01
Asthma                  1.707427e-01
dtype: float64
```

Conclusion: Only the Children and Timely Admission factors have a p value below 0.05. The rest of the variables in the data set do not significantly affect the dependent variable.

#Residuals **-** *Residual is the difference between the observed value and predicted value from our dataset This type of plot is often used to assess whether a linear regression model is fit for our dataset. Evenly distributed residual points between predicted and actual values indicate linear relationship between the dependent and independent variables.*

#Creating predicted values and residuals and adding the two columns to the table

clean_data['Initial_days_predict'] = model_data.predict(x)

clean_data['residual'] = model_data.resid

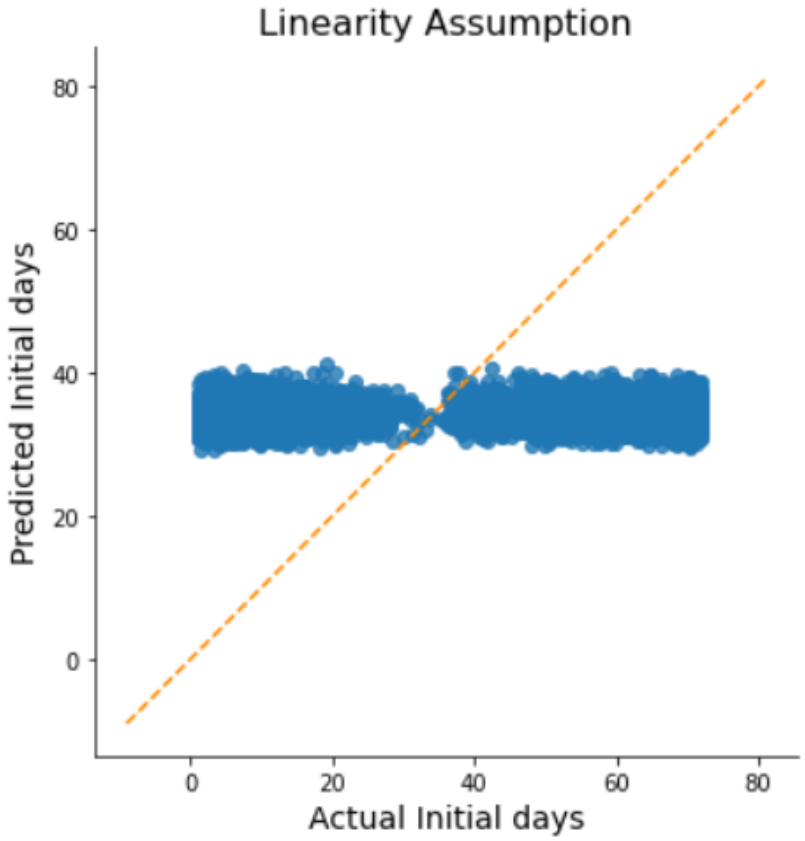
clean_data.head()

Initial_days_predict	residual
35.628467	-25.042697
33.648863	-18.519301
34.240386	-29.468209
36.197541	-34.482662
36.516865	-35.262058

 # Creating a scatter plot of residuals:

 # Plotting the observed vs predicted values

```
sns.lmplot(x='Initial_days', y='Initial_days_predict', data=clean_data, fit_reg=False, size=5)

# Plotting the diagonal line
line_coords = np.arange(clean_data[['Initial_days', 'Initial_days_predict']].min().min()-10,
                        clean_data[['Initial_days', 'Initial_days_predict']].max().max()+10)
plt.plot(line_coords, line_coords,  # X and y points
         color='darkorange', linestyle='--')

plt.ylabel('Predicted Initial days', fontsize=14)
plt.xlabel('Actual Initial days', fontsize=14)
plt.title('Linearity Assumption', fontsize=16)
plt.show()
```



The predicted days do not corelate to the actual days. Predicted days numbers do not have a linear distribution across the plot.

 # Looking for normal distribution of the residual values:

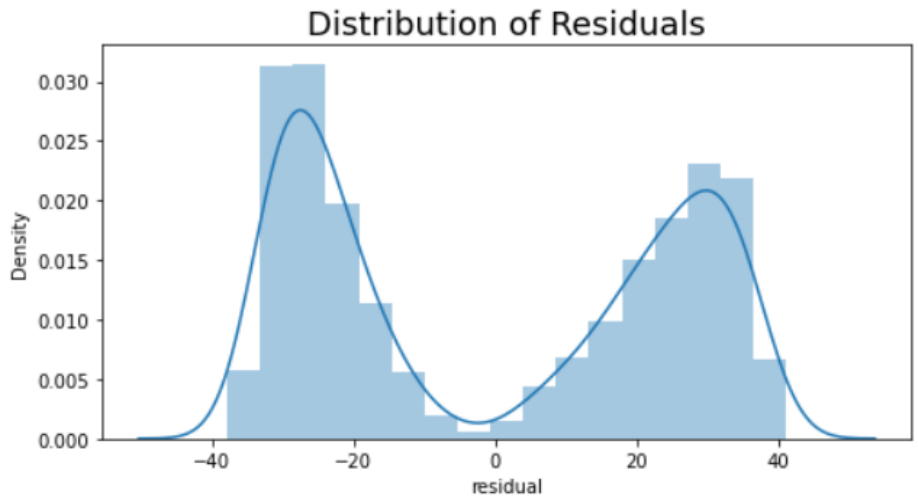
Plotting the residuals distribution

plt.subplots(figsize=(8, 4))

plt.title('Distribution of Residuals', fontsize=18)

sns.distplot(clean_data['residual'])

plt.show()



We can see that compared to normal distribution the residuals are skewed to two peaks. The data is not normally distributed as was the indication from the histogram of initial days in the univariant analysis,

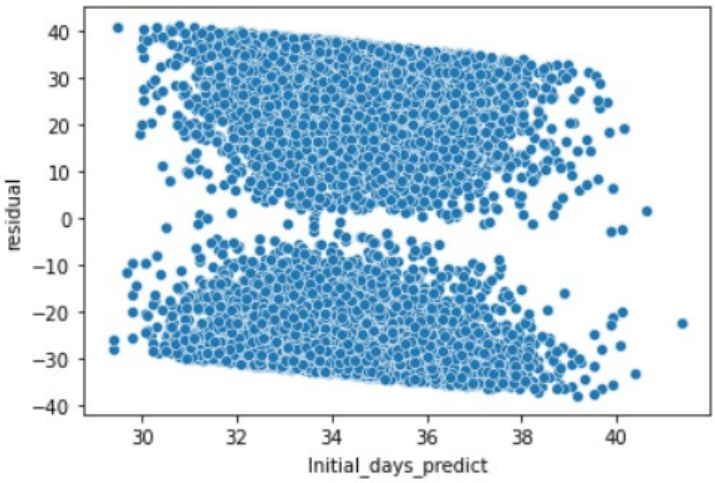
Performing statistical analysis on the residuals: looking for a p_value. P value above 0.05 means normal distribution.

```
from statsmodels.stats.diagnostic import normal_ad

# Performing the test on the residuals

p_value = normal_ad(clean_data['residual'])[1]

print('p-value from the test Anderson-Darling test below 0.05 generally means non-normal:', p_value)

# Reporting the normality of the residuals

if p_value < 0.05:

    print('Residuals are not normally distributed')

else:

    print('Residuals are normally distributed')
```

p-value from the test Anderson-Darling below 0.05 generally means non-normal: 0.0
Conclusion: Residuals are not normally distributed

Plotting residuals vs predicted values will show how many correct predictions the model made. The values near 0 are the closest to correct values.

```
In [19]: # residual plot
         sns.scatterplot(x='Initial_days_predict', y='residual', data=clean_data)
         plt.show()
```



Our model was not able to correctly predict most of the actual values.

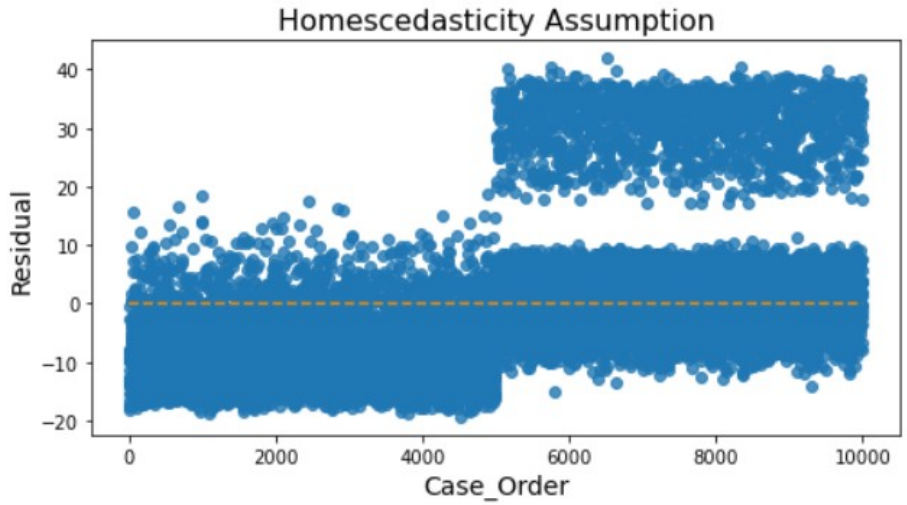
<u># Homoscedasticity</u>

#The variation in the errors across variables is similar.

#Heteroscedasticity, the violation of homoscedasticity, occurs when we do not have an even variance across the error terms. To detect homoscedasticity, we can plot our residual and see if the variance appears to be uniform.

```
# Plotting the residuals

plt.subplots(figsize=(8, 4))

plt.scatter(x=clean_data.index, y=clean_data.residual, alpha=0.8)

plt.plot(np.repeat(0, len(clean_data.index)+2), color='darkorange', linestyle='--')


plt.ylabel('Residual', fontsize=14)

plt.xlabel('Case_Order', fontsize=14)

plt.title('Homescedasticity Assumption', fontsize=16)
```

plt.show()



Conclusion: There appears to be two groups in the dataset that distributed differently. There appears to be Heteroscedasticity across the data.

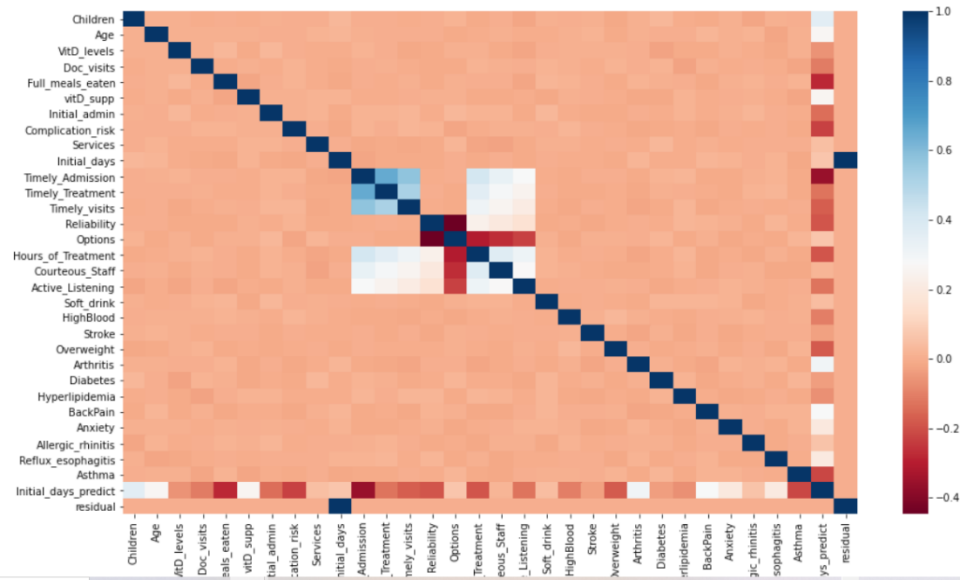
#MultiColiniarity- is there linear relationship between the independent variables. The higher the r 2 value between the variables the bigger is the correlation. We are looking for some relationship but R2 higher than 0.8 will be suspect of multicollinearity and will have to be addressed by reducing the data.

 # Checking for multicolinearity via correlation

corr=clean_data.corr()

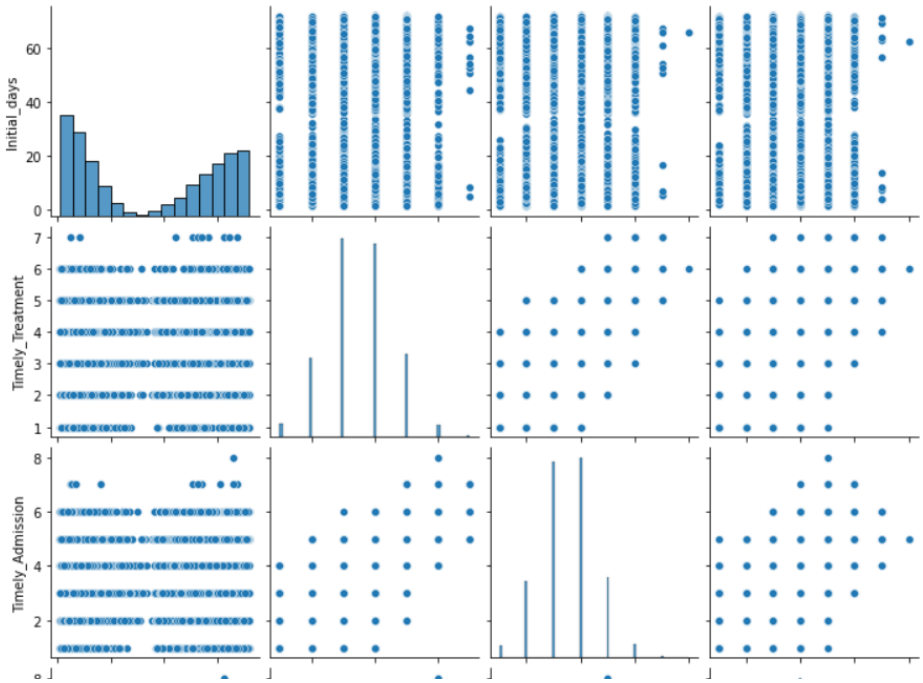
plt.figure(figsize=(16,9))

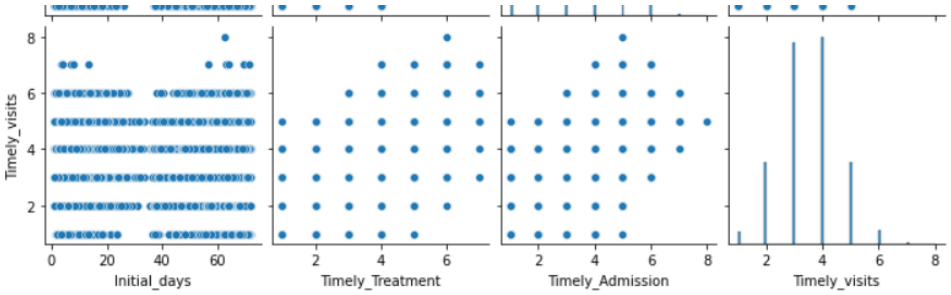
sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, cmap='RdBu')



 # Isolating the independent variables that appear to have some relationship between themselves.

```
sns.pairplot(clean_data[[ 'Initial_days', 'Timely_Treatment', 'Timely_Admission', 'Timely_visits']])

plt.show()
```





There appears to be collinearity between Survey questions: Timely Admission, Timely visits, Timely Treatment. I can reduce the timely items and leave only one of them since it seems they behave identically.

#Looking for multicollinearity with variance inflation factor. We want to reduce the observations that have high collinearity. Another way to detect multicollinearity is by using a metric known as the variance inflation factor (VIF), which measures the correlation and strength of correlation between the explanatory variables in a regression model.

 # Varience inflation factor, large VIF means high multicollinearity

from statsmodels.stats.outliers_influence import variance_inflation_factor

vif=pd.DataFrame()

vif['VIF']=[variance_inflation_factor(x.values, i) for i in range(x.shape[1])]

vif['Features']=x.columns

Vif

	VIF	Features
0	1.938440	Children
1	7.531777	Age
2	51.048970	VitD_levels
3	21.541513	Doc_visits
4	1.989348	Full_meals_eaten
5	1.404381	vitD_supp
6	8.350817	Initial_admin
7	9.140250	Complication_risk
8	4.947672	Services
9	27.906976	Timely_Admission
10	23.609713	Timely_Treatment
11	19.997951	Timely_visits
12	14.251840	Reliability
13	14.041364	Options
14	18.779182	Hours_of_Treatment
15	16.237946	Courteous_Staff
16	14.456076	Active_Listening
17	1.351376	Soft_drink
18	1.692634	HighBlood
19	1.249508	Stroke
20	3.415066	Overweight
21	1.555731	Arthritis
22	1.380371	Diabetes
23	1.510000	Hyperlipidemia
24	1.699859	BackPain
25	1.474097	Anxiety
26	1.649945	Allergic_rhinitis
27	1.701676	Reflux_esophagitis
28	1.409300	Asthma

Conclusion: Variables with VIF values higher than 10 are not going to be included in the analysis. VitD levels, Docs visits, Reliability, Active listening, timely admission, timely visits, can be reduced from the next analysis.

Results summary of the initial model validation:

Conclusion: The model is not valid as it does not hold any of multiple linear regression assumptions.

- Through the process of validating this model I could determine that the R square value of the model is extremely low- looking at the adjusted value of 0. 1% indicating very noisy high variability data.
- There is heteroscedasticity in the data.
- There is no normal distribution of residuals in the data.
- There is no linearity between the independent variables.
- Observing the p values for the independent variables only Children and Timely admission have values below alpha of 0.05. Those factors should be included in the next analysis.
- There was some multicollinearity between independent variables. Looking at the VIF values that determine multilinearity we can isolate values higher than 10. VitD levels, Docs visits, Reliability, Active listening, timely visits may be excluded in the next analysis, according to the correlation matrix they all behave like Timely admissions

Reduced model:

Based on the conclusion from previous model, I will attempt to build a reduced model with the same data, not including independent variables that were determined to have low p value- in the analysis. The purpose of reduced model is to make it more accurate. I will include in this analysis variables with lowest p values from the previous model. I will exclude variables that showed high multicollinearity through VIF values. Unfortunately following a variable selection process, none of the selected variables is categorical data. -Timely admission is ordinal discrete quantitate data and Children is quantitative discrete data type

```
model_rev= model_rev=ols('Initial_days~Children+Timely_Admission', data=clean_data).fit()

print(model_rev.summary()) print(model_rev.summary())
```

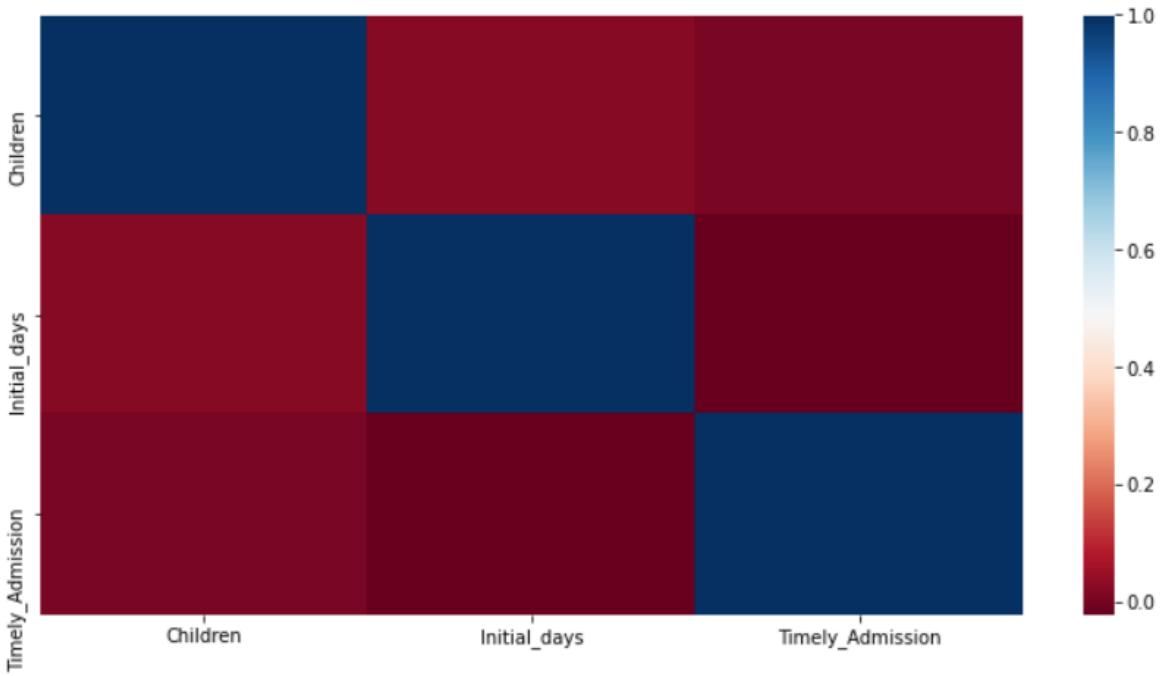
```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Initial_days   R-squared:                       0.001
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     5.015
Date:                Sun, 10 Jul 2022   Prob (F-statistic):            0.00665
Time:                        14:17:17   Log-Likelihood:                -46883.
No. Observations:               10000   AIC:                         9.377e+04
Df Residuals:                    9997   BIC:                         9.379e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept         35.8822      0.968     37.061      0.000      33.984      37.780
Children           0.2738      0.122      2.252      0.024       0.035       0.512
Timely_Admission  -0.5687      0.255     -2.231      0.026      -1.068      -0.069
==============================================================================
Omnibus:                    41245.176   Durbin-Watson:                   0.160
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1285.499
Skew:                           0.071   Prob(JB):                    7.20e-280
Kurtosis:                       1.249   Cond. No.                         16.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

E. **Validation and Analysis of Reduced multiple linear regression model data analysis:**

Checking for multicollinearity to make sure that the columns I have chosen do not have high correlation values.

 # checking for multicolinearity via correlation

```
corr=x.corr()

plt.figure(figsize=(12, 6))

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, cmap='RdBu')
```



No Collinearity is indicated.

Looking at model's statistics:

print('R-square:', model_rev.rsquared)

`R-square: 0.0010023109063076463`

R square result –0.1% of the data variation explained by the independent variables chosen.

When looking at the adjusted R square it is still at 0.1%. This is the same result as for the initial model.

print('F-pvalue:', model_rev.f_pvalue)

`F-pvalue: 0.006653823955992259`

The F pvalue is below 0.05, thus we can reject the null hypothesis, there is some correlation between independent variables combined and the outcome. This is improvement from the initial model and indicates we have chosen the variables that have a correlation to out dependent.

#Looking at ecoefficiencies

print(model_rev.params)

Intercept 35.882234

Children 0.273766

Timely_Admission -0.568682

dtype: float64

Looking at p_values

print(model_rev.pvalues)

```
Intercept           7.757493e-282
Children             2.433020e-02
Timely_Admission     2.567716e-02
dtype: float64
```

Children and Timely Admission have p values below 0.05 just like in the initial model, thus the null hypothesis can be rejected.

Model Validation:

- Checking for linearity. Looking if the predictor variables in the regression have a straight-line relationship with the observed-actual variable.
- Test the assumption of normality and homoscedasticity of our model by looking at residuals. Normally distributed residuals indicate Residual is the difference between the observed value and predicted value from our dataset.

#Making model prediction and adding residuals column to the reduced data set:

x['Initial_days_predict'] = model_rev.predict(x)

x['residual'] = model_rev.resid

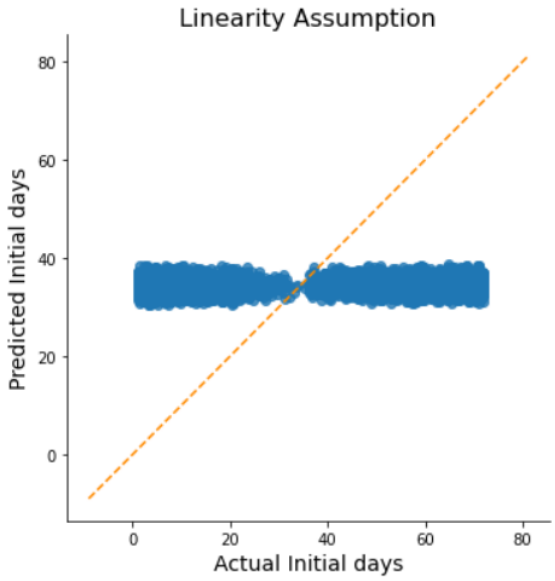
x.columns

```
Index(['Children', 'Initial_days', 'Timely_Admission',
       'Initial_days_predict', 'residual'],
      dtype='object')
```

Columns: predicted Initial days predict and residual were added to the table .

 # Plotting the observed vs predicted values checking for linearity

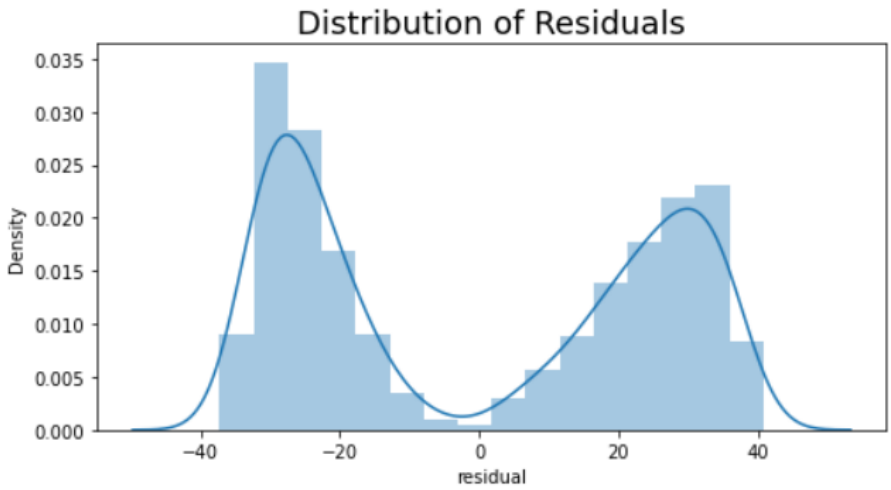
```
sns.lmplot(x='Initial_days', y='Initial_days_predict', data=x, fit_reg=False, size=5)


# Plotting the diagonal line
line_coords = np.arange(x[['Initial_days', 'Initial_days_predict']].min().min()-10,
                        x[['Initial_days', 'Initial_days_predict']].max().max()+10)
plt.plot(line_coords, line_coords,  # X and y points
         color='darkorange', linestyle='--')


plt.ylabel('Predicted Initial days', fontsize=14)
plt.xlabel('Actual Initial days', fontsize=14)
plt.title('Linearity Assumption', fontsize=16)
plt.show()
```



There is no linear relationship between observed and predicted values, same as in the initial model.

Plotting the residuals distribution to check for normality. To make valid inferences from the regression, the residuals of the regression should follow a normal distribution. Non-normality in the residuals means that the amount of error in the model is not consistent across the full range of observed data and can make for a poor prediction model.

```
plt.subplots(figsize=(8, 4))
plt.title('Distribution of Residuals', fontsize=18)
sns.distplot(x['residual'])
plt.show()
```



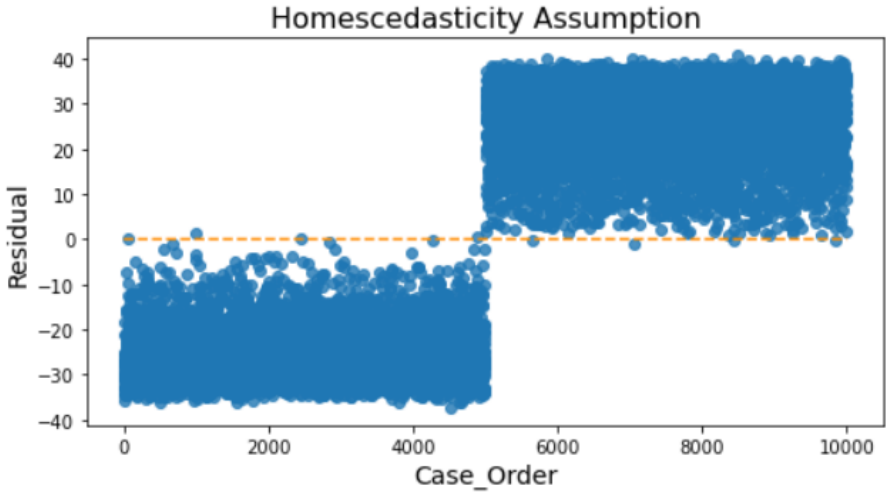
The distribution of the residuals is not normal.

```
from statsmodels.stats.diagnostic import normal_ad


# Performing the test on the residuals
p_value = normal_ad(x['residual'])[1]
print('p-value from the test Anderson-Darling test below 0.05 generally means non-normal:',
    p_value)
# Reporting the normality of the residuals
if p_value < 0.05:
   print('Residuals are not normally distributed')
else:
   print ('Residuals are normally distributed')
```

p-value from the test Anderson-Darling test below 0.05 generally means non-normal distribution of the data : 0.0
Supporting of the conclusion that Residuals are not normally distributed

```
# Plotting the residuals to test homoscedasticity. Residuals are equally distributed, or tend to
    bunch together at some values, and at other values, spread far apart.
```

```
plt.subplots(figsize=(8, 4))

plt.scatter(x=x.index, y=x.residual, alpha=0.8)

plt.plot(np.repeat(0, len(x.index)+2), color='darkorange', linestyle='--')


plt.ylabel('Residual', fontsize=14)

plt.xlabel('Case_Order', fontsize=14)

plt.title('Homescedasticity Assumption', fontsize=16)

plt.show()
```



There is considerable heteroscedasticity in the data. The residual values bunch together at two distinct groups.

Plotting residuals vs predictive values:

```
print(x['residual'].mean())
```

```
Mean= -3.438
```

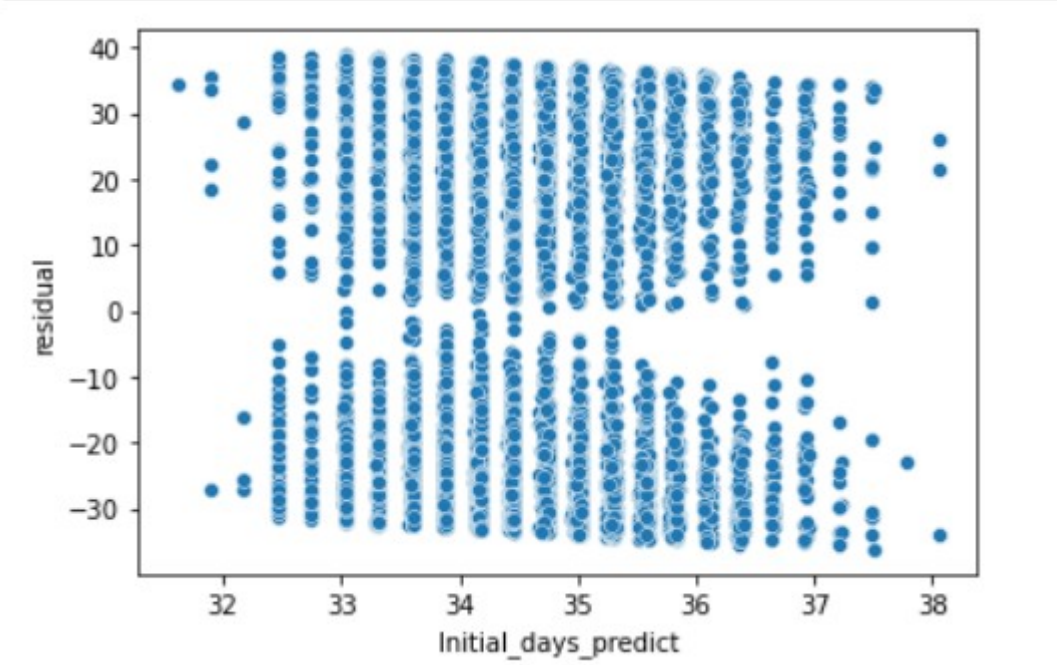
Expected mean to be 0

Residuals= Observed -Predicted

> Creating a residual plot for the reduced model: positive values for the residual (on the y-axis) mean the prediction was too low, and negative values mean the prediction was too high; 0 means the guess was correct (ref 6).

```
# Residual plot:

sns.scatterplot(x='Initial_days_predict', y='residual', data=x)
```

plt.show()



Very few correct predictions in the model.

Table with Predictive values vs Observed values and Residuals:

```
In [46]: print(x[['Initial_days', 'Initial_days_predict', 'residual']].head(10))
```

	Initial_days	Initial_days_predict	residual
0	10.585770	34.449955	-23.864185
1	15.129562	34.997487	-19.867925
2	4.772177	35.566169	-30.793992
3	1.714879	34.176189	-32.461310
4	1.254807	35.018637	-33.763830
5	5.957250	34.428806	-28.471556
6	9.058210	33.607507	-24.549297
7	14.228019	37.229915	-23.001896
8	6.180339	34.176189	-27.995850
9	1.632554	33.586358	-31.953804

Conclusion:

Conclusion: The reduced model is not valid as it does not hold some of multiple linear regression assumptions.

- I was not able to improve the reliability of the model after reduction. Through the process of validating this model I could determine that the R square value of the model is extremely low- looking at the adjusted value of 0.1% indicating very noisy high variability data. The model will not be able to accurately predict outcomes.
- Based on the residual plots both models are not robust in predicting the correct values.
- There is no normal distribution of residuals in the data.
- There is no linearity between the independent variables.
- There is heteroscedasticity in the data.
- Observing the p values for the independent variables; Children and Timely admission have correlation with the dependent variable, those variables were selected based on the analysis of the initial model.
- There was no multicollinearity in the model. Based on the initial model analysis I have eliminated the colinear independent variables.

Part V: Data Summary and Implications

F. 1. Findings and the assumptions:

- **Regression equation for the reduced model:**

  R-square: 0.001=0.1%

  y=35.48+0.275*Children-0.56*Timely Admission

Unfortunately, this model did not present a lot of independent variables with statistically significant relationships to the outcome variable. Two variables that have p values below the significance threshold of 0.05 are Timely Admission and Children. When none of the independent factors present, the hospitalization days will be about 35.5. Timely admission may have a reduction of 0.56 days from the hospitalization days if all the other variables are 0 and timely admission rating was 1, and Children may contribute 0.27-day increase in hospitalization for 1 child.

- This model cannot be used for outcome prediction on this dataset for the research question we were trying to answer. Only two independent variables in the dataset had significant correlation with the dependent variable. The model did not hold the multi regression assumptions even after the data reduction. No linearity, heteroscedasticity, non-normal distribution of the residuals equals weak prediction model.
- According to the analysis of both models it looks like none of the independent factors have linear correlation with the number of hospitalization days.

As a result, based on this model, I was not able to answer the question which factors contribute to prolonged hospitalization. I got some indication for the significance of number of children and timely admissions in relation to the hospitalization days, however as I can not trust the model that does not hold the assumptions true, I cannot make conclusions at this time.

2. Recommendations:

Based on the results of this model I would recommend collecting more data from the hospitals. Increasing the sample size of the patients and expending the observations regarding the chronic conditions of the patients and the reason and the way they were admitted to the hospital. We also could run a model on an alternative question such as is there a link between different health conditions, for example is person with Diabetes more likely to be Overweight and have back problems and then try to link that to the number of hospitalization days. Models that are not based on linear relationships can be used to establish if there is any relationships between factors as well.

References:

1. Complete Dissertation by Statistics Solutions. Assumptions of Multiple Linear Regression - Statistics Solutions
2. Introduction to Python for Data Analysis. Chapter 8: Simple Linear regression. 8. Simple Linear Regression — Basic Analytics in Python (sfu.ca)
3. Multiple Linear Regression Using Python and Scikit-learn (analyticsvidhya.com)
4. Statology by Zach Bobbitt : https://www.statology.org/python-guides/
5. http://faculty.washington.edu/otoomet/machinelearning-py/predictions-model-goodness.html
6. Qualtrics: https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/

Third party code

7. Multiple Linear Regression using Python by Amrutha K; https://www.analyticsvidhya.com/blog/2022/03/multiple-linear-regression-using-python/
8. Multi Linear Regression using Python by Rafi Atha: https://medium.com/swlh/multi-linear-regression-using-python-44bd0d10082d