**PERFORMANCE ASSESSMENT:**

**D208 – Predictive Modeling**

Michael Lawson

Western Governors University

D208 – Predictive Modeling

Keiona Middleton

02/21/2022

Michael C. Lawson | Student ID: #001479557 | Program: MS Data Analytics (5/1/21) |

Program Mentor: Emil Stoica | 859-265-1888 (EST) | [mlaws77@wgu.edu](mailto:mlaws77@wgu.edu)

**Part I: Research Question**

A.  Describe the purpose of this data analysis by doing the following:

1.  Summarize **one** research question that is relevant to a real-world organizational situation captured in the data set you have selected and that you will answer using multiple regression.

Can the number of days a patient will stay on their initial visit be predicted by the data?

2.  Define the objectives or goals of the data analysis. Ensure that your objectives or goals are reasonable within the scope of the data dictionary and are represented in the available data.

The objective of the data analysis is to provide a linear regression model to predict how many days a patient will be staying on an initial visit given the data provided. The hospital will be able to predict things like how many beds will be available soon and which patients will most likely be staying long term. The objective will be to see if health conditions or answers to the survey questions or other factors in the data contribute to a predictive model of initial days at the hospital.

**Part II: Method Justification**

B.  Describe multiple regression methods by doing the following:

1.  Summarize the assumptions of a multiple regression model.

Multiple regression is possible when there is a linear relationship between a dependent (reaction) variable and multiple independent (explanatory) variables. The independent variables can not be too highly correlated causing multicollinearity. The observations for multiple regression must be independently and randomly selected from the population. Residuals should be normally distributed with a mean of zero. The model should have more explanatory power with an increase in variables.

2.  Describe the benefits of using the tool(s) you have chosen (i.e., Python, R, or both) in support of various phases of the analysis.

Both Python and R are great tools for data analysis. Python is the tool chosen for this task., Python uses simple lines of code to import all the tools needed to perform the same tasks built into R. While the additional code may seem like a disadvantage, Python is much more flexible for various uses and has thousands of packages that do so much more than statistics. Python can import and read a wider variety of file types than R. Python is faster at handling large multidimensional arrays using the numpy library and can create beautiful visualizations using packages like Matplotlib and Seaborn. Python is more powerful for data wrangling and web scraping in the future if obtaining more data would be helpful.

3.  Explain why multiple regression is an appropriate technique to analyze the research question summarized in Part I.

Linear regression is appropriate when the response variable is continuous and there is a linear relationship between explanatory variables and the response variable. Adding more than one variable to the regression model creates a better fit for the model. Too many variables can result in overfitting the data. The variable “Initial\_days” is the dependent variable for the question of whether the number of days can be predicted for a patients first visit to the hospital. “Initial\_days” is a continuous numerical variable.

**Part III: Data Preparation**

C.  Summarize the data preparation process for multiple regression analysis by doing the following:

1.  Describe your data preparation goals and the data manipulations that will be used to achieve the goals.

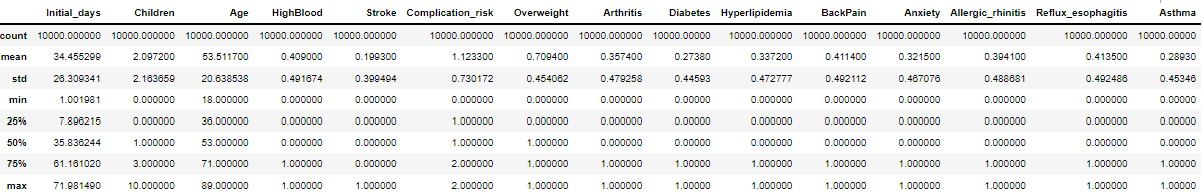
The data preparation goals will be to clean the data, visualize the data, try to find a relationship with the response variable “Initial\_days”, verify a relationship statistically, reduce the variables, run a multiple linear regression model on all the variables selected, reduce the variables using the regression statistics, and then run a regression model on the final set of variables. The data will be cleaned by treating missing values, treating outliers, and checking the data for mistakes such as spelling errors that would create outliers. A gross group of variables will be selected form the data that will exclude variables that obviously wouldn’t contribute to answering the question.

The data will be visualized using scatterplots, boxplots, histograms, etc…The goal of the visualizations is to see how normal the distribution of each variable is, and to see if any relationship can be seen with the response variable before the model is run. Before making any decisions on reducing the data with graphs, the variation inflation factor (VIF) will be calculated on various groups of variables in the data to check for multicollinearity. VIF will help reduce the data further and prepare for principal component analysis (PCA). PCA will help see which variables have the most statistical effect on the data. It will help determine which variables to keep after running recursive feature elimination (RFE). The correlation of the remaining variables will be checked to understand the PCA and RFE results even further. A regression model will be run on all the variables that remained after RFE and PCA. The p-values of the explanatory variables and their relationships will be used to make the final selection of variables. A final linear regression model will be created along with visualizations of the model. The end goal is to find a significant explanation to predict the response variable.

2.  Discuss the summary statistics, including the target variable and all predictor variables that you will need to gather from the data set to answer the research question.

A new data frame was created to exclude any obviously unneeded information in relation to the question. The following variables were selected:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Dtype** | **Quantative/Qualative Dtypes** | **Example** |
| Initial\_days | float64 | Continuous Ratio | 15.129562, 4.772177 |
| Gender | Object | Nominal Categorical | "Male", "Female", "Nonbinary" |
| Initial\_admin | Object | Nominal Categorical | "Emergency Admission" |
| Services | Object | Nominal Categorical | "CT\_Scan", "MRI" |
| Children | int64 | Discrete ratio | 1,2,3,4 |
| Age | int64 | Discrete ratio | 21,56,87 |
| HighBlood | Object | binary categorical | Yes, No |
| Stroke | Object | binary categorical | Yes, No |
| Complication\_risk | Object | ordinal categorical | Yes, No |
| Overweight | Object | binary categorical | Yes, No |
| Arthritis | Object | binary categorical | Yes, No |
| Diabetes | Object | binary categorical | Yes, No |
| Hyperlipidemia | Object | binary categorical | Yes, No |
| BackPain | Object | binary categorical | Yes, No |
| Anxiety | Object | binary categorical | Yes, No |
| Allergic\_rhinitis | Object | binary categorical | Yes, No |
| Reflux\_esophagitis | Object | binary categorical | Yes, No |
| Asthma | Object | binary categorical | Yes, No |



The data includes one ordinal categorical variable, “Complication\_risk”, that can be changed to a numeric ranking of 1-3 for the values “Low-High”, respectively. All of the binary “No/Yes” variables will be changed to 0,1. The categorical variables “Gender”, “Services” and “Initial\_admin” will be changed to dtype ‘categorical’ and the data will be compared categorically against other variables that make it to the final selection. The table above lists all the variables initially selected as the starting gross set of data. The response or dependent variable is the first on the list, “Initial\_days”. This variable is continuous and represents the number of days a patient stayed on the initial visit. Patients stayed between 1 and 72 days, with a mean average of 34 days and a median average of 36 days.

3.  Explain the steps used to prepare the data for the analysis, including the annotated code.

1. Load Python libraries appropriate for data visualization and regression.
2. Load the data using read.csv()
3. Examine the header to see what column names and the values they contain using .head().
4. Examine the shape, dtype, and all column names using .info().
5. Change nominal categorical variables “Gender”, ”Services”, “Marital” and “Initial\_admin” to dtype ‘category’.
6. Change the ordinal categorical variable “Complication\_risk” to ranking numerical values.
7. Rename the survey columns to their respective topic to make them easier to understand.
8. Convert all binary categorical columns from “No,Yes” to “0,1”.
9. If null values exist, treat them.
10. Treat outliers as they are found in visualizations.
11. Create histplots of the four continuous variables in the data set: “Initial\_days”, “Income”, “TotalCharge” and “Additional\_charges”. This step shows distribution of the continuous variables and helped determine if the correct question was being asked. “Initial\_days” is the most normally distributed, so it was used for this analysis.
12. Rather than delete any data, select data needed to create a new data frame of the variables that are most likely to influence the response variable “Initial\_days”.
13. Review the mean, median and quartiles of the data gross data set selected.
14. Create histograms of the “Initial\_days” of every explanatory variable.
15. Perform VIF on groups of variables to test for and drop columns with multicollinearity to help prevent overfitting in the model.
16. Perform PCA to examine which variables have the most effect on the data set.
17. Create a covariance matrix and keep the PCA results above 1.
18. Perform recursive feature extraction (RFE).
19. Compare PCA results and RFE to make a decision on which variables actually have the most effect on the response variable.
20. Check the Pearson correlation of the selected variables and their relationships with each other and the response variable.
21. Use scatterplots with lineplots to visualize different combinations of the remaining variables.
22. Use ordinary least squares (OLS) to estimate the known parameters in the linear regression model.
23. Consider dropping variables with a p-value over 0.05. Try OLS on different combinations of variables to make sure the columns being considered for elimination don’t react differently.
24. Decide on the final variables for the finished model.
25. Create visualizations of the final model, using different combinations of the variables if needed.

***Annotated Code***

*# import all possible packages useful for multiple linear regression*

**import** pandas **as** pd

**import** numpy **as** np

**import** sklearn

**from** pandas **import** Series, DataFrame

**from** pylab **import** rcParams

**from** sklearn **import** preprocessing

**from** sklearn.preprocessing **import** StandardScaler

**from** sklearn.datasets **import** load\_iris

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn **import** metrics

**from** sklearn.metrics **import** classification\_report, confusion\_matrix

**import** matplotlib.pyplot **as** plt

plt**.**rc("font", size **=** 14)

**%matplotlib** inline

**from** scipy **import** stats

**import** seaborn **as** sns

sns**.**set(style**=**"white")

sns**.**set(style**=**"whitegrid", color\_codes**=True**)

**from** sklearn.feature\_selection **import** RFE

**from** sklearn.ensemble **import** AdaBoostRegressor

**from** sklearn.datasets **import** load\_boston

**from** numpy **import** array

**from** statsmodels.stats.outliers\_influence **import** variance\_inflation\_factor

**from** statsmodels.formula.api **import** ols

**import** statsmodels.api **as** sm

**import** os

os**.**getcwd()

**from** sklearn **import** linear\_model

**from** sklearn.metrics **import** mean\_squared\_error

**import** warnings

warnings**.**filterwarnings('ignore')

*#Import data set from hard drive*

med **=** pd**.**read\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D208\medical\_clean.csv", skiprows**=**0, delimiter**=**",")

*#examine the list of variables, their data type and the shape of the data*

med**.**info()

A picture containing table

Description automatically generated

*#print the header*

med**.**head()

Graphical user interface, text, application, email

Description automatically generated

*#change the dtype of categorical values to 'categorical'*

med["Gender"] **=** med["Gender"]**.**astype('category')

med['Services'] **=** med['Services']**.**astype('category')

med['Initial\_admin'] **=** med['Initial\_admin']**.**astype('category')

med['Marital'] **=** med['Marital']**.**astype('category')

*#change the values of the ordinal categorical variable to numeric ranks*

med['Complication\_risk']**.**unique()

med['Complication\_risk'] **=** med['Complication\_risk']**.**map({'Low':0,'Medium':1,'High':2})

*#change all binary categorical values to 0,1*

med['Soft\_drink'] **=** med['Soft\_drink']**.**map({'No':0, "Yes":1})

med['HighBlood'] **=** med['HighBlood']**.**map({'No':0, "Yes":1})

med['Stroke'] **=** med['Stroke']**.**map({'No':0, "Yes":1})

med['Overweight'] **=** med['Overweight']**.**map({'No':0, "Yes":1})

med['Arthritis'] **=** med['Arthritis']**.**map({'No':0, "Yes":1})

med['Diabetes'] **=** med['Diabetes']**.**map({'No':0, "Yes":1})

med['Hyperlipidemia'] **=** med['Hyperlipidemia']**.**map({'No':0, "Yes":1})

med['BackPain'] **=** med['BackPain']**.**map({'No':0, "Yes":1})

med['Anxiety'] **=** med['Anxiety']**.**map({'No':0, "Yes":1})

med['Allergic\_rhinitis'] **=** med['Allergic\_rhinitis']**.**map({'No':0, "Yes":1})

med['Reflux\_esophagitis'] **=** med['Reflux\_esophagitis']**.**map({'No':0, "Yes":1})

med['Asthma'] **=** med['Asthma']**.**map({'No':0, "Yes":1})

*#make the survey columns easier to understand*

med**.**rename(columns **=**{'Item1':'S1\_Timely\_admission', 'Item2':'S2\_Timely\_Treatment', 'Item3':'S3\_Timely\_Visits', 'Item4':'S4\_Reliability', 'Item5':'S5\_Options', 'Item6':'S6\_Hours\_of\_Treatment', 'Item7':'S7\_Courteous\_staff', 'Item8':'S8\_Dr\_Actively\_Listened'}, inplace**=True**)

*#visualize the distribution of each continuous (float64) variable in the data set. "Initial\_days" is the most significant to the question, but as you can see is also the most normal distribution of the continuous variables.*

sns**.**histplot(data**=**med, x**=**"Initial\_days", bins**=**10)

*Chart, histogram

Description automatically generated*

sns**.**histplot(data**=**med, x**=**"Income", bins**=**10)

*Chart, histogram

Description automatically generated*

sns**.**histplot(data**=**med, x**=**"TotalCharge", bins**=**10)

*Chart, histogram

Description automatically generated*

sns**.**histplot(data**=**med, x**=**"Additional\_charges", bins**=**10)

*Chart, histogram

Description automatically generated*

*#Gross selection of data before PCA*

da\_df **=** med[["Initial\_days", "Gender","Initial\_admin", "Services", "Children", "Age", "HighBlood", "Stroke", "Complication\_risk", "Overweight", "Arthritis", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Allergic\_rhinitis", "Reflux\_esophagitis", "Asthma", "S1\_Timely\_admission", "S2\_Timely\_Treatment", "S3\_Timely\_Visits", "S4\_Reliability", "S5\_Options", "S6\_Hours\_of\_Treatment", "S7\_Courteous\_staff", "S8\_Dr\_Actively\_Listened" ]]

*#look at the data types and what variables remain*

da\_df**.**info()

Table

Description automatically generated

*#mean, median, and qaurtiles*

da\_df**.**describe()

Table

Description automatically generated

*#Create histograms of the number of initial days for each explanatory variable*

sns**.**displot(data**=**da\_df, x**=**"Initial\_days", col**=**"Initial\_admin", col\_wrap**=**3, bins**=**10)

plt**.**show()

*Chart, histogram

Description automatically generated*

sns**.**displot(data**=**da\_df, x**=**"Initial\_days", col**=**"Gender", col\_wrap**=**3, bins**=**10)

plt**.**show()

*Chart, histogram

Description automatically generated*

sns**.**displot(data**=**da\_df, x**=**"Initial\_days", col**=**"Services", col\_wrap**=**4, bins**=**10)

plt**.**show()

*Chart, histogram

Description automatically generated*

*# VIF for demographic data. Adding gender columns returned an error when mixed with marital*

X **=** da\_df[['Children', 'Age']]

vif\_demo\_data **=** pd**.**DataFrame()

vif\_demo\_data['Demographic'] **=** X**.**columns

vif\_demo\_data["VIF"] **=** [variance\_inflation\_factor(X**.**values, i)

**for** i **in** range(len(X**.**columns))]

print(vif\_demo\_data)



X2 **=** da\_df[['S1\_Timely\_admission', 'S2\_Timely\_Treatment','S3\_Timely\_Visits','S4\_Reliability','S5\_Options', 'S6\_Hours\_of\_Treatment','S7\_Courteous\_staff','S8\_Dr\_Actively\_Listened']]

vif\_surv\_data **=** pd**.**DataFrame()

vif\_surv\_data['Demographic'] **=** X2**.**columns

vif\_surv\_data["VIF"] **=** [variance\_inflation\_factor(X2**.**values, i)

**for** i **in** range(len(X2**.**columns))]

print(vif\_surv\_data)

Text

Description automatically generated

*#Drop columns with a VIF > 10 leaving only 1 to represent the data*

da\_df **=** da\_df**.**drop(columns**=**['S1\_Timely\_admission', 'S2\_Timely\_Treatment','S3\_Timely\_Visits','S4\_Reliability','S5\_Options', 'S6\_Hours\_of\_Treatment','S7\_Courteous\_staff','S8\_Dr\_Actively\_Listened'])

*#Look over remaining variables and check dtypes*

da\_df**.**info()

Graphical user interface

Description automatically generated with low confidence

*#Gather a sample to start checking statistical data*

np**.**random**.**seed(944)

data **=** da\_df**.**sample(100)

*#Perform PCA*

**from** sklearn.decomposition **import** PCA

med\_pca **=** da\_df[["Initial\_days","Children", "Age", "HighBlood", "Stroke", "Complication\_risk", "Overweight", "Arthritis", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Allergic\_rhinitis", "Reflux\_esophagitis", "Asthma"]]

med\_normalized **=** (med\_pca**-**med\_pca**.**mean())**/**med\_pca**.**std()

pca **=** PCA(n\_components **=** med\_pca**.**shape[1])

pca**.**fit(med\_normalized)

loadings **=** pd**.**DataFrame(pca**.**components\_**.**T,

columns **=** ['PCA1', 'PCA2', 'PCA3', 'PCA4','PCA5','PCA6', 'PCA7', 'PCA8','PCA9','PCA10','PCA11','PCA12','PCA13','PCA14', 'PCA15'],

index**=**med\_normalized**.**columns)

loadings

*Table

Description automatically generated*

*#Create a covariance matrix to decide which PCAs to keep*

cov\_matrix **=** np**.**dot(med\_normalized**.**T, med\_normalized)**/**med\_pca**.**shape[0]

eigenvalues **=** [np**.**dot(eigenvector**.**T, np**.**dot(cov\_matrix, eigenvector)) **for** eigenvector **in** pca**.**components\_]

plt**.**plot(eigenvalues, 'ro-')

plt**.**xlabel('Number of Components')

plt**.**ylabel('Eigenvalues')

plt**.**show()

*Chart, line chart

Description automatically generated*

*#Standardize the data so that all the variables are uniform in comparison*

x **=** data[["Children", "Age", "HighBlood", "Stroke", "Complication\_risk", "Overweight", "Arthritis", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Allergic\_rhinitis", "Reflux\_esophagitis", "Asthma"]]

y **=** data["Initial\_days"]

x\_data**=**x

target**=**y

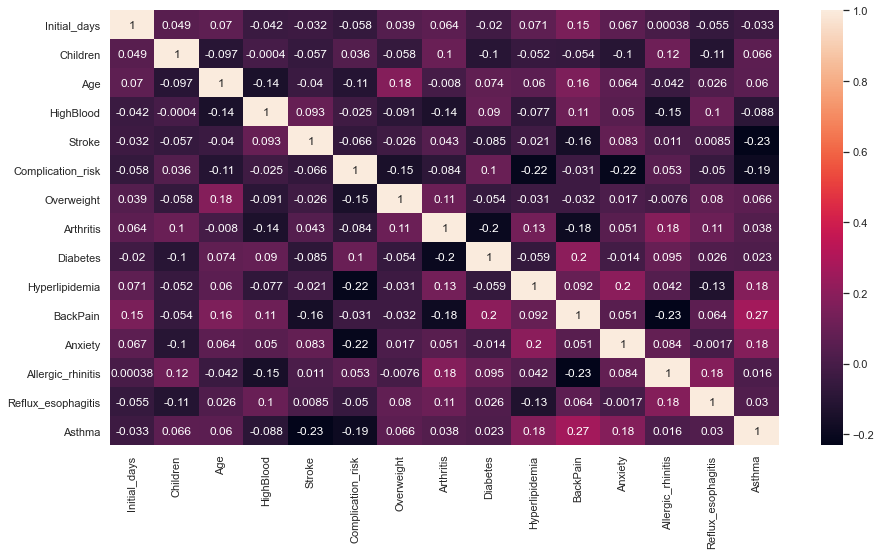
scaled\_data**=**preprocessing**.**scale(x\_data)

*#heatmap to view correlation*

sns**.**set(rc **=** {'figure.figsize':(15,8)})

sns**.**heatmap(data**.**corr(), annot**=True**)

plt**.**show()

**

*#Check Pearson correlation*

data**.**corr(method**=**'pearson')

*Graphical user interface, table

Description automatically generated*

*#Create bivariate visualizations of the data*

sns**.**scatterplot(data**=**data, x**=**'Age', y**=**"Initial\_days", color**=**"Red")

sns**.**lineplot(data**=**data, x**=**'Age', y**=**"Initial\_days", ci**=None**,color**=**"Orange")

**

sns**.**scatterplot(data**=**data, x**=**'Children', y**=**"Initial\_days", color**=**"purple")

sns**.**lineplot(data**=**data, x**=**'Children', y**=**"Initial\_days", ci**=None**,color**=**"Blue")

*Chart, line chart

Description automatically generated*

sns**.**scatterplot(data**=**data, x**=**'Stroke', y**=**"Initial\_days", color**=**"aqua")

sns**.**lineplot(data**=**data, x**=**'Stroke', y**=**"Initial\_days", ci**=None**,color**=**"pink")

*Chart, histogram

Description automatically generated*

sns**.**scatterplot(data**=**data, x**=**'Complication\_risk', y**=**"Initial\_days", color**=**"green")

sns**.**lineplot(data**=**data, x**=**'Complication\_risk', y**=**"Initial\_days", ci**=None**,color**=**"yellow")

*Chart, line chart, scatter chart

Description automatically generated*

sns**.**scatterplot(data**=**data, x**=**'Overweight', y**=**"Initial\_days", color**=**"blue")

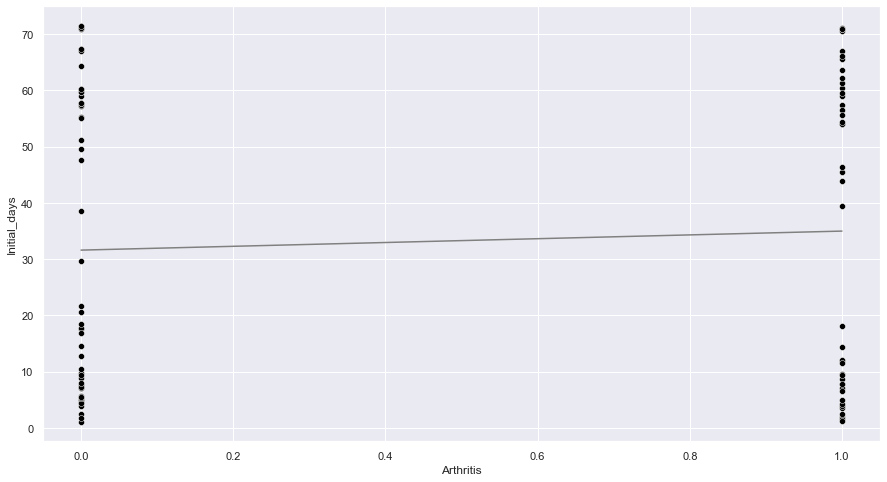
sns**.**lineplot(data**=**data, x**=**'Overweight', y**=**"Initial\_days", ci**=None**,color**=**"purple")

*Chart, histogram

Description automatically generated*

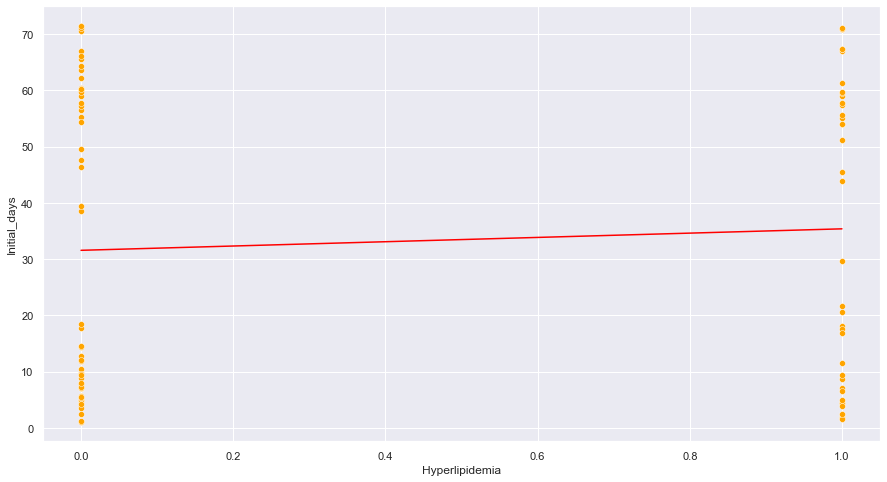
sns**.**scatterplot(data**=**data, x**=**'Arthritis', y**=**"Initial\_days", color**=**"black")

sns**.**lineplot(data**=**data, x**=**'Arthritis', y**=**"Initial\_days", ci**=None**,color**=**"grey")

**

sns**.**scatterplot(data**=**data, x**=**'Hyperlipidemia', y**=**"Initial\_days", color**=**"orange")

sns**.**lineplot(data**=**data, x**=**'Hyperlipidemia', y**=**"Initial\_days", ci**=None**,color**=**"red")

**

sns**.**scatterplot(data**=**data, x**=**'Diabetes', y**=**"Initial\_days", color**=**"aqua")

sns**.**lineplot(data**=**data, x**=**'Diabetes', y**=**"Initial\_days", ci**=None**,color**=**"black")

*Chart, histogram

Description automatically generated*

sns**.**scatterplot(data**=**data, x**=**'Allergic\_rhinitis', y**=**"Initial\_days", color**=**"grey")

sns**.**lineplot(data**=**data, x**=**'Allergic\_rhinitis', y**=**"Initial\_days", ci**=None**,color**=**"black")

*Chart, histogram

Description automatically generated*

sns**.**scatterplot(data**=**data, x**=**'Reflux\_esophagitis', y**=**"Initial\_days", color**=**"red")

sns**.**lineplot(data**=**data, x**=**'Reflux\_esophagitis', y**=**"Initial\_days", ci**=None**,color**=**"grey")

*Chart, histogram

Description automatically generated*

sns**.**scatterplot(data**=**data, x**=**'BackPain', y**=**"Initial\_days", color**=**"violet")

sns**.**lineplot(data**=**data, x**=**'BackPain', y**=**"Initial\_days", ci**=None**,color**=**"purple")

*Chart, line chart

Description automatically generated*

*#Perform Recursive Feature Extraction to select which variables have the most statistical significance to "Initial\_days"*

np**.**random**.**seed(944)

x **=** data[["Children", "Age", "HighBlood", "Stroke", "Complication\_risk", "Overweight", "Arthritis", "Diabetes", "Hyperlipidemia", "BackPain", "Anxiety", "Allergic\_rhinitis", "Reflux\_esophagitis", "Asthma"]]

y **=** data["Initial\_days"]

estimator **=** AdaBoostRegressor(random\_state**=**0, n\_estimators**=**100)

selector **=** RFE(estimator, n\_features\_to\_select**=**5, step**=**1)

selector **=** selector**.**fit(x, y)

filter **=** selector**.**support\_

ranking **=** selector**.**ranking\_

print("Mask data: ", filter)

print("Ranking: ", ranking)

Mask data: [ True True False False True False False False False True False False

True False]

Ranking: [ 1 1 9 7 1 6 10 3 5 1 4 8 1 2]

*#print RFE*

features **=** array(x**.**columns**.**values)

print("All features:")

print(features)

print("Selected features:")

print(features[filter])

All features:

['Children' 'Age' 'HighBlood' 'Stroke' 'Complication\_risk' 'Overweight'

'Arthritis' 'Diabetes' 'Hyperlipidemia' 'BackPain' 'Anxiety'

'Allergic\_rhinitis' 'Reflux\_esophagitis' 'Asthma']

Selected features:

['Children' 'Age' 'Complication\_risk' 'BackPain' 'Reflux\_esophagitis']

*#Standardize the data so that all the variables are uniform in comparison*

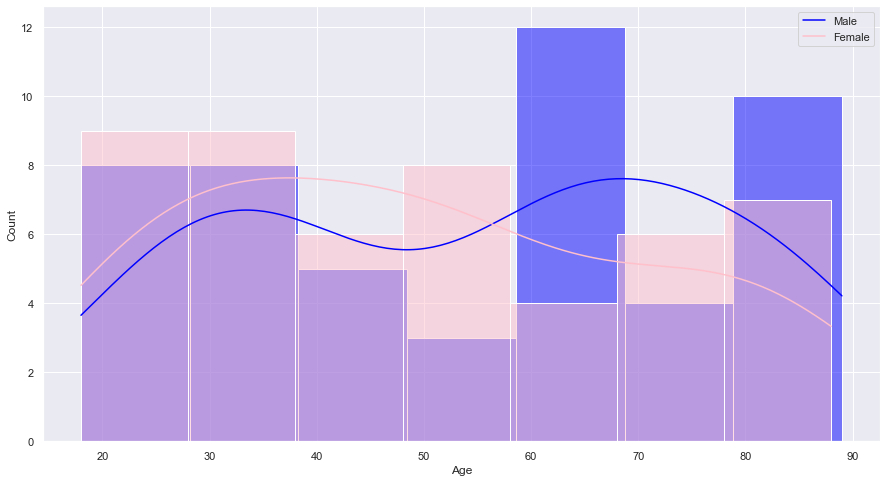
M **=** data[data['Gender']**==**'Male']

F **=** data[data["Gender"]**==**'Female']

sns**.**histplot(data**=**M, x**=**'Age', kde**=True**, color**=**"Blue")

sns**.**histplot(data**=**F, x**=**'Age', kde**=True**, color**=**'Pink')

plt**.**legend(labels**=**["Male", "Female"], bbox\_to\_anchor**=**(1,1))

**

*#Use scatterplot with lineplot to visualize any relationship between Age and Iniital\_days against medical conditions*

BP **=** data[data["HighBlood"]**==**1]

stroke**=** data[data['Stroke']**==**1]

hyper **=** data[data['Hyperlipidemia']**==**1]

sns**.**scatterplot(data**=**BP, x**=**'Age', y**=**"Initial\_days", color**=**"Red")

sns**.**scatterplot(data**=**stroke, x**=**'Age', y**=**"Initial\_days", color**=**"Purple")

sns**.**scatterplot(data**=**hyper, x**=**'Age', y**=**'Initial\_days', color**=**"Orange")

sns**.**lineplot(data**=**BP, x**=**'Age', y**=**"Initial\_days", ci**=None**,color**=**"Red")

sns**.**lineplot(data**=**stroke, x**=**'Age', y**=**"Initial\_days", ci**=None**, color**=**"Purple")

sns**.**lineplot(data**=**hyper, x**=**'Age', y**=**'Initial\_days', ci**=None**, color**=**"Orange")

plt**.**legend(labels**=**["Hypertensive","Stroke", "Hyperlipidemia"], bbox\_to\_anchor**=**(1,1))

*Chart, line chart

Description automatically generated*

complication **=** data[data['Complication\_risk']**==**1]

obese **=** data[data['Overweight']**==**1]

intra **=** data[data['Services']**==**'Intravenous']

sns**.**scatterplot(data**=**complication, x**=**'Age', y**=**"Initial\_days", color**=**"Blue")

sns**.**scatterplot(data**=**obese, x**=**'Age', y**=**'Initial\_days', color**=**"Magenta")

sns**.**scatterplot(data**=**intra, x**=**'Age', y**=**'Initial\_days', color**=**'Orange')

sns**.**lineplot(data**=**complication, x**=**'Age', y**=**"Initial\_days", ci**=None**, color**=**"Blue")

sns**.**lineplot(data**=**obese, x**=**'Age', y**=**"Initial\_days", ci**=None**, color**=**"Magenta")

sns**.**lineplot(data**=**intra, x**=**'Age', y**=**'Initial\_days', ci**=None**, color**=**'Orange')

plt**.**legend(labels**=**['Complication','Obese', 'Intravenous'], bbox\_to\_anchor**=**(1,1))

**

*#Use scatterplot with lineplot to visualize any relationship between Age and Iniital\_days against type of admission*

ER **=** data[data["Initial\_admin"]**==**'Emergency Admission']

OBS **=** data[data['Initial\_admin']**==**'Observation Admission']

Elect **=** data[data['Initial\_admin']**==**'Elective Admission']

sns**.**scatterplot(data**=**ER, x**=**'Age', y**=**"Initial\_days", color**=**"Red")

sns**.**scatterplot(data**=**OBS, x**=**'Age', y**=**"Initial\_days", color**=**"Green")

sns**.**scatterplot(data**=**Elect, x**=**'Age', y**=**"Initial\_days", color**=**"Blue")

sns**.**lineplot(data**=**ER, x**=**'Age', y**=**"Initial\_days", ci**=None**,color**=**"Red")

sns**.**lineplot(data**=**OBS, x**=**'Age', y**=**"Initial\_days", ci**=None**, color**=**"Green")

sns**.**lineplot(data**=**Elect, x**=**'Age', y**=**"Initial\_days", ci**=None**, color**=**"Blue")

plt**.**legend(labels**=**["Emergency","Observation", "Elective"], bbox\_to\_anchor**=**(1,1))

**

*#Use scatterplot with lineplot to visualize any relationship between # of Children and Iniital\_days against type of admission*

ER **=** data[data["Initial\_admin"]**==**'Emergency Admission']

OBS **=** data[data['Initial\_admin']**==**'Observation Admission']

Elect **=** data[data['Initial\_admin']**==**'Elective Admission']

sns**.**scatterplot(data**=**ER, x**=**'Children', y**=**"Initial\_days", color**=**"Red")

sns**.**scatterplot(data**=**OBS, x**=**'Children', y**=**"Initial\_days", color**=**"Green")

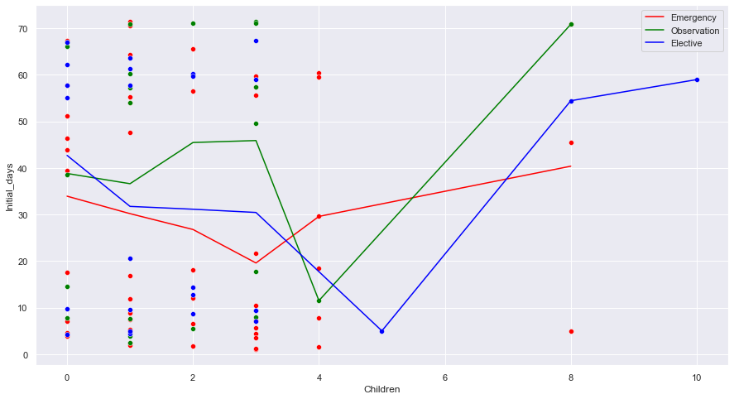
sns**.**scatterplot(data**=**Elect, x**=**'Children', y**=**"Initial\_days", color**=**"Blue")

sns**.**lineplot(data**=**ER, x**=**'Children', y**=**"Initial\_days", ci**=None**,color**=**"Red")

sns**.**lineplot(data**=**OBS, x**=**'Children', y**=**"Initial\_days", ci**=None**, color**=**"Green")

sns**.**lineplot(data**=**Elect, x**=**'Children', y**=**"Initial\_days", ci**=None**, color**=**"Blue")

plt**.**legend(labels**=**["Emergency","Observation", "Elective"], bbox\_to\_anchor**=**(1,1))

**

*#First Ordinary Least Squares Model*

mdl\_A **=** ols("Initial\_days ~ Initial\_admin + Gender + HighBlood + Stroke + Reflux\_esophagitis + Services + Children + Age + Allergic\_rhinitis + Asthma + Anxiety + BackPain + Complication\_risk + Overweight + Arthritis + Diabetes + Hyperlipidemia + 0", data**=**da\_df)**.**fit()

print(mdl\_A**.**summary())

Table

Description automatically generated

*#Second Ordinary Least Squares Model*

mdl\_B **=** ols("Initial\_days ~ Initial\_admin + Children + Age + Initial\_admin:Age + Initial\_admin:Children + Age:Children + 0", data**=**data)**.**fit()

print(mdl\_B**.**summary())

Table

Description automatically generated

*#Third Ordinary Least Squares Model. Eliminated variables with high p-value*

mdl\_C **=** ols("Initial\_days ~ Initial\_admin + Age + 0", data**=**data)**.**fit()

print(mdl\_C**.**summary())

Table

Description automatically generated

*#Modeling the strongest explanatory variable to the response variable alone*

mdl\_D **=** ols("Initial\_days ~ Initial\_admin + Children + 0", data**=**data)**.**fit()

print(mdl\_D**.**summary())

Table

Description automatically generated

*#Find the mean of Initial\_days of each category for variable “Initial\_admin”*

admin\_summary\_stats **=** data**.**groupby("Initial\_admin")["Initial\_days"]**.**mean()

print(admin\_summary\_stats)

Text

Description automatically generated

Emerg **=** data[data['Initial\_admin']**==**'Emergency Admission']

Observ **=** data[data['Initial\_admin']**==**'Observation Admission']

Elect **=** data[data['Initial\_admin']**==**'Elective Admission']

sns**.**regplot(x**=**"Children", y**=**"Initial\_days", data**=**Emerg, ci**=None**, color**=**"violet")

sns**.**regplot(x**=**"Children", y**=**"Initial\_days", data**=**Observ, ci**=None**, color**=**"blue")

sns**.**regplot(x**=**"Children", y**=**"Initial\_days", data**=**Elect, ci**=None**, color**=**"black")

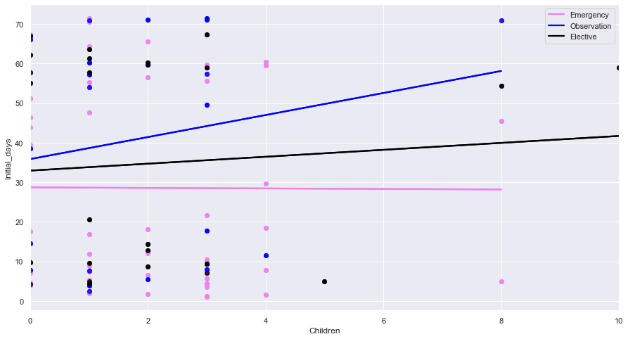
sns**.**regplot(x**=**"Children", y**=**"Initial\_days", data**=**Emerg, ci**=None**, color**=**"violet")

sns**.**regplot(x**=**"Children", y**=**"Initial\_days", data**=**Observ, ci**=None**, color**=**"blue")

sns**.**regplot(x**=**"Children", y**=**"Initial\_days", data**=**Elect, ci**=None**, color**=**"black")

plt**.**legend(labels**=**['Emergency','Observation', 'Elective'])

plt**.**show()

**

Emerg **=** data[data['Initial\_admin']**==**'Emergency Admission']

Observ **=** data[data['Initial\_admin']**==**'Observation Admission']

Elect **=** data[data['Initial\_admin']**==**'Elective Admission']

sns**.**regplot(x**=**"Age", y**=**"Initial\_days", data**=**Emerg, ci**=None**, color**=**"violet")

sns**.**regplot(x**=**"Age", y**=**"Initial\_days", data**=**Observ, ci**=None**, color**=**"blue")

sns**.**regplot(x**=**"Age", y**=**"Initial\_days", data**=**Elect, ci**=None**, color**=**"black")

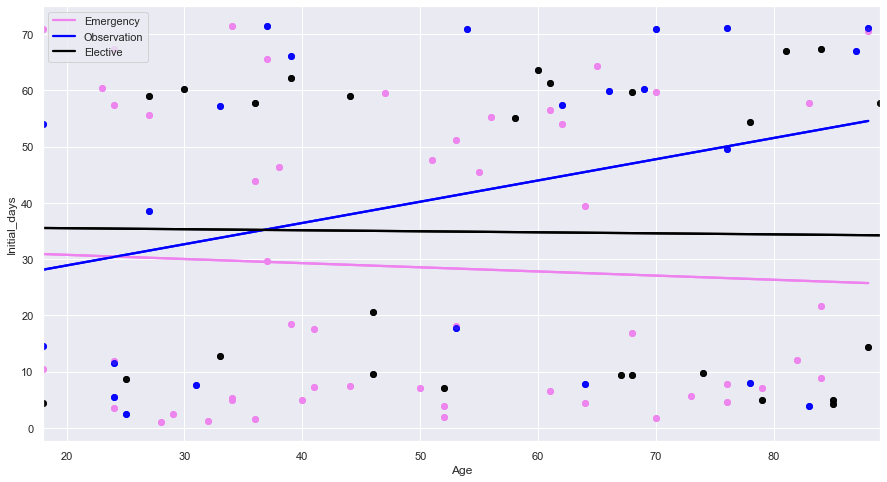
sns**.**regplot(x**=**"Age", y**=**"Initial\_days", data**=**Emerg, ci**=None**, color**=**"violet")

sns**.**regplot(x**=**"Age", y**=**"Initial\_days", data**=**Observ, ci**=None**, color**=**"blue")

sns**.**regplot(x**=**"Age", y**=**"Initial\_days", data**=**Elect, ci**=None**, color**=**"black")

plt**.**legend(labels**=**['Emergency','Observation', 'Elective'])

plt**.**show()

**

*#Residual plots*

Emerg **=** data[data['Initial\_admin']**==**'Emergency Admission']

Observ **=** data[data['Initial\_admin']**==**'Observation Admission']

Elect **=** data[data['Initial\_admin']**==**'Elective Admission']

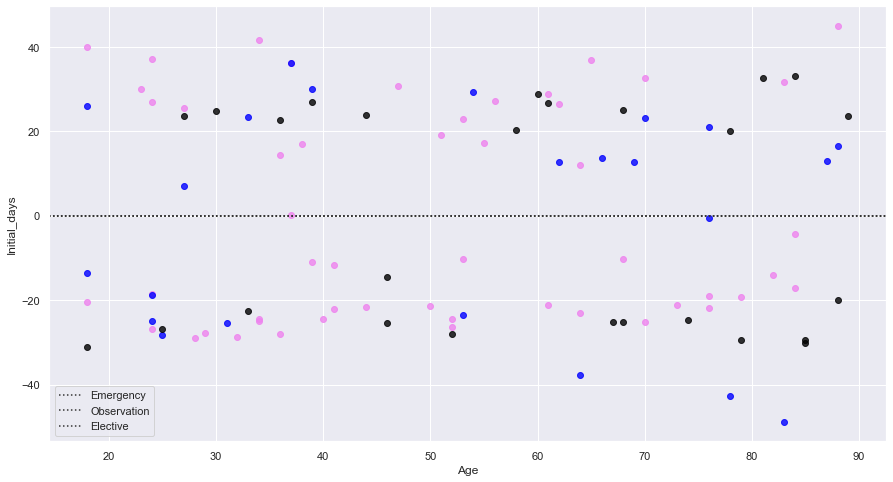
sns**.**residplot(x**=**"Age", y**=**"Initial\_days", data**=**Emerg, color**=**"violet")

sns**.**residplot(x**=**"Age", y**=**"Initial\_days", data**=**Observ, color**=**"blue")

sns**.**residplot(x**=**"Age", y**=**"Initial\_days", data**=**Elect, color**=**"black")

plt**.**legend(labels**=**['Emergency','Observation', 'Elective'])

plt**.**show()

**

*#Residual plots*

Emerg **=** data[data['Initial\_admin']**==**'Emergency Admission']

Observ **=** data[data['Initial\_admin']**==**'Observation Admission']

Elect **=** data[data['Initial\_admin']**==**'Elective Admission']

sns**.**residplot(x**=**"Children", y**=**"Initial\_days", data**=**Emerg, color**=**"violet")

sns**.**residplot(x**=**"Children", y**=**"Initial\_days", data**=**Observ, color**=**"blue")

sns**.**residplot(x**=**"Children", y**=**"Initial\_days", data**=**Elect, color**=**"black")

plt**.**legend(labels**=**['Emergency','Observation', 'Elective'])

plt**.**show()

*A screenshot of a computer

Description automatically generated with medium confidence*

*#Export prepared data*

data**.**to\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D208\medical\_prepared\_D208.csv")

4.  Generate univariate and bivariate visualizations of the distributions of variables in the cleaned data set. Include the target variable in your bivariate visualizations.

**Univariate:**

Chart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generatedChart, histogram

Description automatically generated

**Bivariate:**

Chart, scatter chart

Description automatically generatedChart, line chart

Description automatically generated

A picture containing chart

Description automatically generatedA picture containing chart

Description automatically generated

Chart, line chart

Description automatically generatedA picture containing chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generated with low confidence Chart

Description automatically generated with low confidence

Chart

Description automatically generated with low confidenceChart, line chart, scatter chart

Description automatically generated

5.  Provide a copy of the prepared data set.

**Attached as separate .csv file:**

**data.to\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D208\medical\_prepared\_D208.csv")**

**Part IV: Model Comparison and Analysis**

D.  Compare an initial and a reduced multiple regression model by doing the following:

1.  Construct an initial multiple regression model from all predictors that were identified in Part C2.

Table

Description automatically generated

2.  Justify a statistically based variable selection procedure and a model evaluation metric to reduce the initial model in a way that aligns with the research question.

The p-value should be lower that 0.05 on variables that are strong influencers, otherwise the influence of the variable could be due to random chance. The model in D1 shows that the variable “Children” and the categories under “Initial\_admin” are the only variables we can realiably say are strong not just due to chance, but p-values are known to swing wildly according to how many, and which variables are included in the model.

PCA, RFE, and correlation were all examined to make sure the best variables were chosen for the final model. The outputs are shown below in E1. In PCA, “Initial\_admin” still came up as the variable influenceing the response variable the most, closely followed by “Age” and “Children”. RFE also picked “Age” and “Children” in the top 5 most important variables in reference to the response variable “Initial\_days”. The final model will use the response variable “Initial\_days” and the explanatory variables “Initial\_admin” and “Children”.

3.  Provide a reduced multiple regression model that includes both categorical and continuous variables.

“Initial\_admin” is the categorical value chosen for the final model. Age is a continuous variable rounded to a whole number in this data set. “Children” is a discrete numerical variable. Each variable and their relationships have been modeled.

Table

Description automatically generated

A couple extra models were run just out of curiosity of how the p-values would react.

Table

Description automatically generated

Table

Description automatically generated

The R-squared and Adj R-squared of every model was very low. The p-values swung wildy for Age and Children, but “Initial\_admin stayed strong as an influencer of the response variable with a confidence that the relationship is not by random chance. Multicollinearity was removed with RFE and correlation didn’t reveal any highly correlated variables with the response variable. The conclusion is “Initial\_admin” may predict something about the “Initial\_days” a patient will stay, but the remaining variables either are not influencers, or the data is not good. Given the wild spread on the variable “Initial\_days, with a large portion missing in the middle of the data, it appears some data must be missing form the set.

E.  Analyze the data set using your reduced multiple regression model by doing the following:

1.  Explain your data analysis process by comparing the initial and reduced multiple regression models, including the following elements:

•  the logic of the variable selection technique

Correlation was reviewed to make sure no explanatory variables were too highly correlated with the response variable:

Timeline

Description automatically generated

Table

Description automatically generated

Variance inflation factor (VIF) was used to check for multicollinearity, and variables with a VIF higher than 10 were removed.

Text

Description automatically generated

Principal Component Analysis (PCA) revealed the most influential variables in the data set. The higher the value, the more influential the variable is.

A computer screen capture

Description automatically generated with low confidence

A covariance matrix revealed which PCA columns to examine. 8 Eigevalues land above 1.

Chart, line chart

Description automatically generated

•  the model evaluation metric

The R2 value is highest on the model with “Initial\_admin”, “Children” and “Age” together. The adj R2 value is negative, which means that when the original residuals are considered along with the predicted, the explanation toward response is extremely low. The last model with “Initial\_admin” and “Children” has the highest adj R2 value. None of the models have a high enough R2 or adj R2 value to show that the variables used give a strong explanation towards the response. The standard error is smaller across most variables in the initial model with all the variables than in the reduced model. The F-statistic is not significantly different between the initial and the reduced model. A higher log-likelyhood indicats a better fit. The initial model had a log-likelyhood of -46874 and the reduced model of -467. This indicates the reduced model is a better fit.

•  a residual plot

Chart, scatter chart

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

2.  Provide the output and any calculations of the analysis you performed, including the model’s residual error.

Note: The output should include the predictions from the refined model you used to perform the analysis.

All output was shown with code to avoid confusion.

3.  Provide the code used to support the implementation of the multiple regression models.

All code was provided above. The code specifically associated with the final regression model is as follows:

mdl\_D = ols("Initial\_days ~ Initial\_admin + Children + 0",data=data).fit()

print(mdl\_D.summary())

**Part V: Data Summary and Implications**

F.  Summarize your findings and assumptions by doing the following:

1.  Discuss the results of your data analysis, including the following elements:

•  a regression equation for the reduced model

The final model was mdl\_D, which included the response variable “Initial\_days”, and the explanatory variables “Initial\_admin”, and “Children”

The regression formula from the model is:

Y = (33.1761\*26.9053)+(33.1761\*39.634)+(33.1761\*0.7637)+0

•  an interpretation of coefficients of the statistically significant variables of the model

The coefficients reveal that at x=0, there are more observational hospital visits than elective visits, and more elective visits than emergency.

•  the statistical and practical significance of the model

The visualization of the model appears to reveal that emergency visits don’t effect the number of days a patient stays; however, elective and observational visits result in longer hospital stays the more children a patient has.The findings are insignificant though. This data could never be presented for any decisions to stakeholders at the hospital, because the high p-values and low R2 and adj R2 values show that children provide little explanation for the response variable.

•  the limitations of the data analysis

The data was limited to mostly binary variables. The response variable “Initial\_days” appeared to be nearly empty between 20-45 days, meaning the dataset is probably missing tuples or there was an issue with the data collection process. Only “Initial\_admin” stayed statisitically strong through the entire preparation stage, and so multiple linear regression was difficult. No combination of variables provided strong adj R2 values or significant p-values.

2.  Recommend a course of action based on your results.

The recommendation is in order to answer the question of how to predict how many days a patient will stay on an initial visit, more data is needed. The initial recommendation would be to investigate the gap in initial\_days to see if an explanation for the missing data can be found. Data needs to be collected and added, or this data needs to be thrown out and a new population of data examined. This data has little statistically significant information in relationship to “Initial\_days”, so a prediction with high confidence can not be made in reference to how many days a patient will stay on an initial visit.

**Part VI: Demonstration**

G.  Provide a Panopto video recording that includes all of the following elements:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3c834367-8686-4331-a653-ae4301769734

H.  List the web sources used to acquire data or segments of third-party code to support the application. Ensure the web sources are reliable.

WGU D208 Predictive Modeling Webinar -Episode 1. William Sewell. [*https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william\_sewell\_wgu\_edu/ER\_vJMbYtxJGpxImpZ0DUQcBoVcORYKanFVKNKFcEXkRow?e=trkoWb*](https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william_sewell_wgu_edu/ER_vJMbYtxJGpxImpZ0DUQcBoVcORYKanFVKNKFcEXkRow?e=trkoWb)

WGU D208 Predictive Modeling Webinar -Episode 2. William Sewell. [*https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william\_sewell\_wgu\_edu/Edu1kfXcHXNJpgvLe2FGGfUB-W0IKyXxTVxOPAx3malF\_A?e=xkg2hx*](https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william_sewell_wgu_edu/Edu1kfXcHXNJpgvLe2FGGfUB-W0IKyXxTVxOPAx3malF_A?e=xkg2hx)

WGU D208 Predictive Modeling Webinar -Episode 3. William Sewell. *https://westerngovernorsuniversity-my.sharepoint.com/:p:/g/personal/william\_sewell\_wgu\_edu/ESVjAM1CmBxCpMZ-2Nt-lmAB\_WbiOhLSz8GP4DFFAb5HZg?e=DuKgwl*

sklearn.feature\_selection.RFE. Scikitlearn Developers (2021*)* [*https://scikit-learn.org/stable/modules/generated/sklearn.feature\_selection.RFE.html#sklearn.feature\_selection.RFE*](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html#sklearn.feature_selection.RFE)

Python – How to Draw Confusion Matrix Using Matpllotlib (September 1, 2020) [*https://vitalflux.com/python-draw-confusion-matrix-matplotlib/*](https://vitalflux.com/python-draw-confusion-matrix-matplotlib/)

Introduction to Linear Modeling in Python. DataCamp (2022) *https://campus.datacamp.com/courses/introduction-to-linear-modeling-in-python/exploring-linear-trends?ex=1*

1. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

Massaron, L., & Boschetti, A. (2016). Regression analysis with python: Learn the art of regression analysis with python. Packt Publishing.

J.  Demonstrate professional communication in the content and presentation of your submission.