Course project guidelines

Your assignment for the course project is to formulate and answer a question of your choosing based on one of the following datasets:

1. ClimateWatch historical emissions data: greenhouse gas emissions by U.S. state 1990-present
2. World Happiness Report 2023: indices related to happiness and wellbeing by country 2008-present
3. Any dataset from the class assignments or mini projects

A good question is one that you want to answer. It should be a question with contextual meaning, not a purely technical matter. It should be clear enough to answer, but not so specific or narrow that your analysis is a single line of code. It should require you to do some nontrivial exploratory analysis, descriptive analysis, and possibly some statistical modeling. You aren't required to use any specific methods, but it should take a bit of work to answer the question. There may be multiple answers or approaches to contrast based on different ways of interpreting the question or different ways of analyzing the data. If your question is answerable in under 15 minutes, or your answer only takes a few sentences to explain, the question probably isn't nuanced enough.

Deliverable

Prepare and submit a jupyter notebook that summarizes your work. Your notebook should contain the following sections/contents:

- **Data description**: write up a short summary of the dataset you chose to work with following the conventions introduced in previous assignments. Cover the sampling if applicable and data semantics, but focus on providing high-level context and not technical details; don't report preprocessing steps or describe tabular layouts, etc.
- **Question of interest**: motivate and formulate your question; explain what a satisfactory answer might look like.
- **Data analysis**: provide a walkthrough with commentary of the steps you took to investigate and answer the question. This section can and should include code cells and text cells, but you should try to focus on presenting the analysis clearly by organizing cells according to the high-level steps in your analysis so that it is easy to skim. For example, if you fit a regression model, include formulating the explanatory variable matrix and response, fitting the model, extracting coefficients, and perhaps even visualization all in one cell; don't separate these into 5-6 substeps.
- **Summary of findings**: answer your question by interpreting the results of your analysis, referring back as appropriate. This can be a short paragraph or a bulleted list.

Evaluation

Your work will be evaluated on the following criteria:

1. Thoughtfulness: does your question reflect some thoughtful consideration of the dataset and its nuances, or is it more superficial?
2. Thoroughness: is your analysis an end-to-end exploration, or are there a lot of loose ends or unexplained choices?
3. Mistakes or oversights: is your work free from obvious errors or omissions, or are there mistakes and things you've overlooked?
4. Clarity of write-up: is your report well-organized with commented codes and clear writing, or does it require substantial effort to follow?

Analysis of Life Ladder Based on Quality of Life Metrics

Kaden Nichols and Riley Zamora

Data description

The **World Happiness Report (WHR)** is an annual publication released by the United Nations Sustainable Developement Solutions Network (SDSN) which explains which country is in the best state of happiness and well-being. The data is collected by the Gallup World Poll which uses a standardized set of survey questions to measure various aspects of well-being and happiness. The data set we are looking at in this project contains data on the WHR from 2008-present. There are 11 variables and 2199 observations in the uncleaned raw data. The variables and their defenitions are provided in a table below.

Variable Name	Description
Country name	Name of the country
Year	Year of the observation
Life Ladder	Measure of subjective well-being (see more full definition below)
Log GDP per capita	Logarithm of the country's GDP per capita
Social support	Perceived social support; someone to count on? National average of binary responses (0,1)
Healthy life expectancy at birth	Average number of years of healthy life
Freedom to make life choices	Satisfied with the freedom to make life choices?
Generosity	Residual of regressing national average answer to the question “Have you donated money to a charity in the past month?”
Perceptions of corruption	“Is corruption widespread throughout the government or not” “Is corruption widespread within businesses or not?” The overall perception is just the average of the two 0-or-1 responses.

Variable Name	Description
Positive affect	These measures include responses to the following questions: "Did you smile or laugh a lot yesterday?", "Did you experience enjoyment during a lot of the day yesterday?", and "Did you learn or do something interesting yesterday?"
Negative affect	Average of three negative affect measures in GWP. Worry, sadness, anger for question “Did you experience the following feelings during A LOT OF THE DAY yesterday?"

Life ladder : "Happiness score or subjective well-being (variable name ladder): The survey measure of SWB is from the Jan 20, 2023 release of the Gallup World Poll (GWP) covering years from 2005 to 2022. Unless stated otherwise, it is the national average response to the question of life evaluations. The English wording of the question is “Please imagine a ladder, with steps numbered from 0 at the bottom to 10 at the top. The top of the ladder represents the best possible life for you and the bottom of the ladder represents the worst possible life for you. On which step of the ladder would you say you personally feel you stand at this time?” This measure is also referred to as Cantril life ladder, or just life ladder in our analysis." (Statistical Appendix for World Happiness Report 2023)

Question of interest

With what subset of metrics of the quality of life in nations around the world- GDP, freedom, social support, corruption, life expectancy, etc.- can we best predict the nation's happiness level as measured by our variable called Life Ladder. How does each quality of life metric affect life ladder nations in each continent?

An ideal answer to this question would be to show the overall trend in world happiness, then be broken down to continents happiness levels. We would then estimate life ladder based off the best combination of expanatory variables that are chosen with model selection techniques.

Data analysis

EDA

To get started it is always a good idea to get comfotable with our data. To do this we can perform some exploratory data analysis, which includes visualization, getting dimensions, checking missing data, and gain insight on basic statistical measures of the numeric columns.

In [67]:

```
import pandas as pd
import numpy as np
import statsmodels.api as sm # for regression
#!pip install missingno
import missingno as msno # for missingness
import altair as alt # plotting
alt.renderers.enable('mimetype')
```

```python
import matplotlib.pyplot as plt # plotting
#!pip install pycountry_convert
import pycountry_convert as pc # country to continent

# Read in the data, store as panda df
whr = pd.read_csv('data/whr-2023.csv')

# Check the shape of our dataset
print(whr.shape)

# First few rows
whr.head()
```

```
(2199, 11)
```

Out[67]:

	Country name	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generosity	Perceptions of corruption
0	Afghanistan	2008	3.724	7.350	0.451	50.5	0.718	0.168	0.882
1	Afghanistan	2009	4.402	7.509	0.552	50.8	0.679	0.191	0.850
2	Afghanistan	2010	4.758	7.614	0.539	51.1	0.600	0.121	0.707
3	Afghanistan	2011	3.832	7.581	0.521	51.4	0.496	0.164	0.731
4	Afghanistan	2012	3.783	7.661	0.521	51.7	0.531	0.238	0.776

Great! This allows us to get and overview of our dataset by inspecting the first few rows. There are 11 variables and 2199 observations. Now, we know what we are working with and can start deciding what needs to be cleaned. After running the command `whr.info()` we get the following information that tells us the structure of our data. Formatting into a markdown table helps us better visualize exatly what is going on in the output.

Column	Non-Null Count	Dtype
Country name	2199 non-null	object
year	2199 non-null	int64
Life Ladder	2199 non-null	float64
Log GDP per capita	2179 non-null	float64
Social support	2186 non-null	float64
Healthy life expectancy at birth	2145 non-null	float64
Freedom to make life choices	2166 non-null	float64
Generosity	2126 non-null	float64
Perceptions of corruption	2083 non-null	float64
Positive affect	2175 non-null	float64
Negative affect	2183 non-null	float64

Uh oh, we have some missing values in our data. This can affect model outcomes or skew our data. Lets dig into our numeric columns and check some basic statistical measures using `whr.describe()` so we can get an idea of what is going on.

In [68]:

```
whr.describe()
```

Out[68]:

	year	Life Ladder	Log GDP per capita	Social support	Healthy life expectancy at birth	Freedom to make life choices	Generos
count	2199.000000	2199.000000	2179.000000	2186.000000	2145.000000	2166.000000	2126.000
mean	2014.161437	5.479227	9.389760	0.810681	63.294582	0.747847	0.000
std	4.718736	1.125527	1.153402	0.120953	6.901104	0.140137	0.161
min	2005.000000	1.281000	5.527000	0.228000	6.720000	0.258000	-0.338
25%	2010.000000	4.647000	8.500000	0.747000	59.120000	0.656250	-0.112
50%	2014.000000	5.432000	9.499000	0.836000	65.050000	0.770000	-0.023
75%	2018.000000	6.309500	10.373500	0.905000	68.500000	0.859000	0.092
max	2022.000000	8.019000	11.664000	0.987000	74.475000	0.985000	0.703

From viewing the mean value along with the min, max, and quantiles it is apparent that we are also dealing with some outliers in our data. It is not immidiately known whether these will affect our data but knowing about them can help us to make decisions later on. For now, lets focus on cleaning up the data.

Data Cleaning

To make the data more readable we will clean the variable names. Firstly, we will covert the variables to be lower case which will help with readability. We will then rename some of the variables to be easier to call. Do not worry the column names will still be easy to interpret as readability is our main goal for data cleaning.

In [69]:

```
# Convert to Lowercase
whr.columns = whr.columns.str.lower()

# Rename columns that are too long and hard to call
whr_name = whr.rename(columns={'country name': 'country', 'log gdp per capita':
                               'healthy life expectancy at birth': 'life expecta
                               'freedom to make life choices': 'freedom',
                               'perceptions of corruption': 'corruption'})

# Make sure our data set is updated
whr_name.head()
```

Out[69]:

	country	year	life ladder	log gdp	social support	life expectancy	freedom	generosity	corruption	po
0	Afghanistan	2008	3.724	7.350	0.451	50.5	0.718	0.168	0.882	
1	Afghanistan	2009	4.402	7.509	0.552	50.8	0.679	0.191	0.850	
2	Afghanistan	2010	4.758	7.614	0.539	51.1	0.600	0.121	0.707	
3	Afghanistan	2011	3.832	7.581	0.521	51.4	0.496	0.164	0.731	
4	Afghanistan	2012	3.783	7.661	0.521	51.7	0.531	0.238	0.776	

In our specific question, we are testing how log(gdp), social support, life expectancy, freedom, and corruption affect the life ladder from years 2008-present. To help with visualization, I am going to cohort countries by their continent. This will show us the same global trend while giving us insight on how happy each continent is in general. Fist, we will see how many unique countries are included in the data. This will let us know which countries are included as well as how many. Using a package called `pycountry_convert` we will be able to take each unique country and output a dictionary of country name to coninent. We will use this to then subset the included countries to their continents. I found the documentation for `pycountry_convert` here.

In [70]:

```
# set unique country
unique_countries = whr_name['country'].unique()
len(unique_countries) # 165

# build a function using pycountry_convert
def country_to_continent(country_name):
    country_alpha2 = pc.country_name_to_country_alpha2(country_name)
    country_continent_code = pc.country_alpha2_to_continent_code(country_alpha2)
    country_continent_name = pc.convert_continent_code_to_continent_name(country
    return country_continent_name

# loop through unique_countries to convert and append to dictionary
country_dict = {}
for country in unique_countries:
    if (country == 'Congo (Brazzaville)' or country == 'Congo (Kinshasa)' or
        country == 'Hong Kong S.A.R. of China' or country == 'Kosovo' or
        country == 'Somaliland region' or country == 'State of Palestine' or
        country == 'Taiwan Province of China' or country == 'Turkiye'):
        next
    else:
        continent = country_to_continent(country)
        country_dict[country] = continent

# add in countries that don't work with pycountry_convert
country_dict['Congo (Brazzaville)'] = 'Africa'
country_dict['Congo (Kinshasa)'] = 'Africa'
country_dict['Hong Kong S.A.R. of China'] = 'Asia'
country_dict['Kosovo'] = 'Europe'
country_dict['Somaliland region'] = 'Africa'
country_dict['State of Palestine'] = 'Asia'
country_dict['Taiwan Province of China'] = 'Asia'
country_dict['Turkiye'] = 'Asia'
```

```
# add a new column with continents
whr_name['continent'] = whr_name['country'].map(country_dict)

# view to see
whr_name.head()
```

Out[70]:

	country	year	life ladder	log gdp	social support	life expectancy	freedom	generosity	corruption	po
0	Afghanistan	2008	3.724	7.350	0.451	50.5	0.718	0.168	0.882	
1	Afghanistan	2009	4.402	7.509	0.552	50.8	0.679	0.191	0.850	
2	Afghanistan	2010	4.758	7.614	0.539	51.1	0.600	0.121	0.707	
3	Afghanistan	2011	3.832	7.581	0.521	51.4	0.496	0.164	0.731	
4	Afghanistan	2012	3.783	7.661	0.521	51.7	0.531	0.238	0.776	

Adding this new column will help us visualize our data a little bit better. We are almost done with our data wrangling, all we need to do now is remove the columns that do not pertain to our question of interest. I will remove positive effect and negative effect.

In [71]:

```
# remove columns and create a new data set
whr_interest = whr_name.drop(columns = ['positive affect', 'negative affect'])

# view to see
whr_interest.head()
```

Out[71]:

	country	year	life ladder	log gdp	social support	life expectancy	freedom	generosity	corruption	co
0	Afghanistan	2008	3.724	7.350	0.451	50.5	0.718	0.168	0.882	
1	Afghanistan	2009	4.402	7.509	0.552	50.8	0.679	0.191	0.850	
2	Afghanistan	2010	4.758	7.614	0.539	51.1	0.600	0.121	0.707	
3	Afghanistan	2011	3.832	7.581	0.521	51.4	0.496	0.164	0.731	
4	Afghanistan	2012	3.783	7.661	0.521	51.7	0.531	0.238	0.776	

Handling Missing Values

Lets take a further look into our missing data values and what variables have them. using `whr.isnull().sum()` we obtain the total missing values for each column.

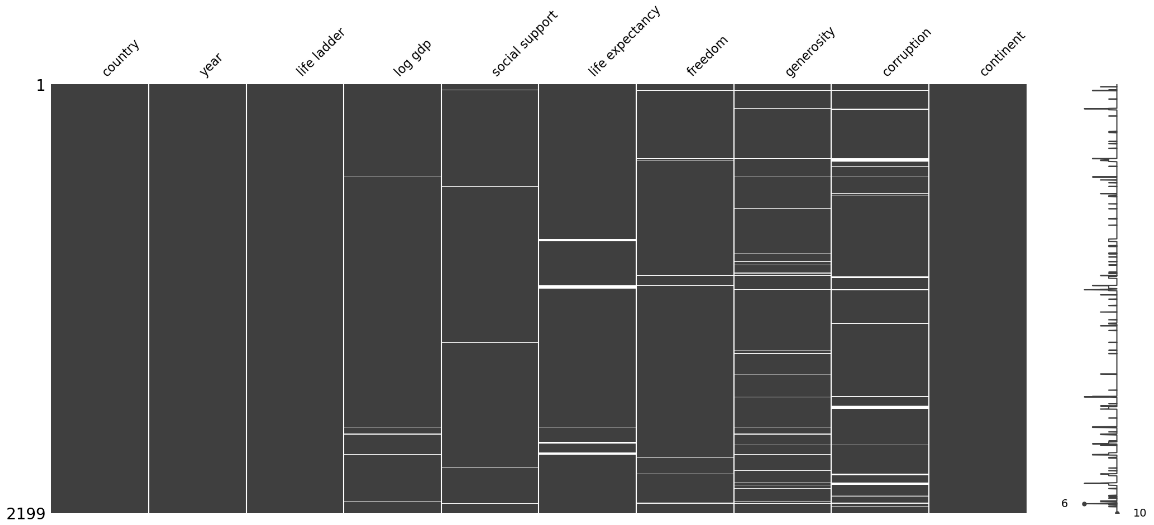
Column	Null Count
Country name	0
year	0
Life Ladder	0
Log GDP per capita	20
Social support	13

Column	Null Count
Healthy life expectancy at birth	54
Freedom to make life choices	33
Generosity	73
Perceptions of corruption	116
Positive affect	24
Negative affect	16

Wow, perceptions of coorruption has 116 missing values! Understanding missing data is important because we need to make certain decisions on how to handle missing data. We have a few options for how we could handle our missing data such as, dropping the columns or rows that have na's, or using imputation to fill the na's. There are a few different imputation techniques we could use to fill the na's such as iterpolation, filling (mean, median, mode), or using backward/forward filling. To choose which method will work best we need to understand how the missing data is spread. To do this I will use the `missingno` library to visualize a missing data matrix.

In [72]:

```
# Visualize missing data spread
msno.matrix(whr_interest)
plt.show()
```

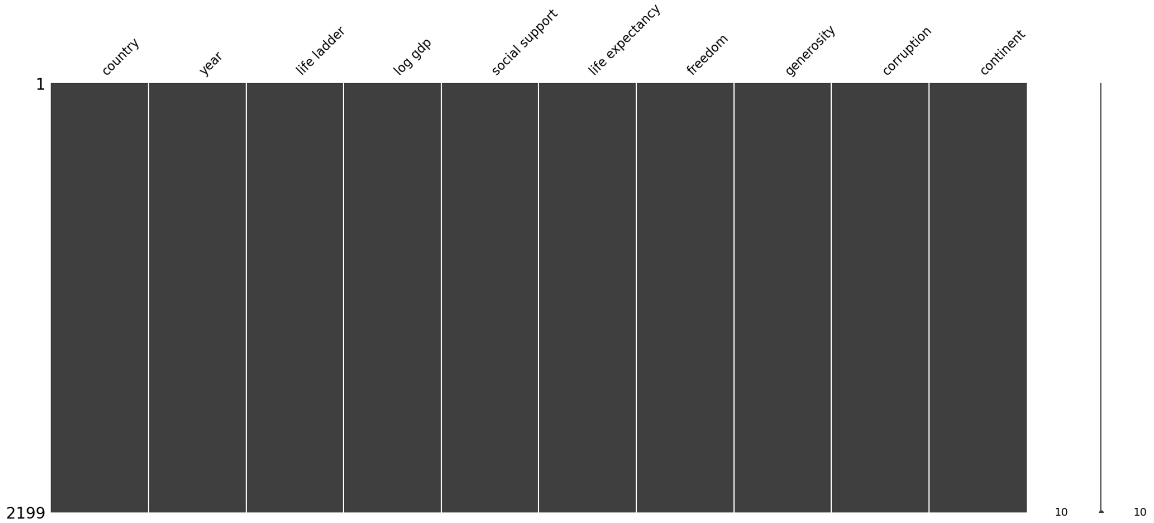


From this figure, we can see how spread our missing values are. In Healthy life expectancy at birth and perceptions of corruption, the thicker white lines show where missing values are clustered. When missing values are densly packed, we should stay away from methods like filling with a central tendency beacuse these methods assume the missing data occurs randomly or are more spread out. Instead we should consider forward/backward filling or using interpolation. We can also consider dropping the na's altogether. Let's try a few different methods and check how they affect our data with `.describe()`

In [73]:

```
# Testing how interpolate affects our data
whr_interpolate = whr_interest.interpolate()
msno.matrix(whr_interpolate)
plt.show()
```

```
whr_interpolate.describe()
```



Out[73]:

	year	life ladder	log gdp	social support	life expectancy	freedom	genero
count	2199.000000	2199.000000	2199.000000	2199.000000	2199.000000	2199.000000	2199.000
mean	2014.161437	5.479227	9.386313	0.810925	63.269249	0.748085	0.000
std	4.718736	1.125527	1.155453	0.120724	6.862399	0.140085	0.159
min	2005.000000	1.281000	5.527000	0.228000	6.720000	0.258000	-0.338
25%	2010.000000	4.647000	8.494500	0.747500	59.200909	0.657000	-0.109
50%	2014.000000	5.432000	9.498000	0.836000	64.960000	0.770000	-0.023
75%	2018.000000	6.309500	10.374000	0.905000	68.345000	0.859875	0.092
max	2022.000000	8.019000	11.664000	0.987000	74.475000	0.985000	0.703

All the na's are filled! Comparing to our descriptive statistics before filling the missing values, we can see that using interpolation did a good job. Looking at the descriptive statistics across the columns in table above vs below, adding in these data points did not drastically affect our std, mean, or quartiles. We would say this was an efficient method for dealing with the missing values. The only column that shows any sort of change is generosity, we belive this is because the data values are so miniscule. Additionally, since this did a sufficient job we see no need to check forward/backward filling.

In [74]:

```
# Show descriptive stats from before imputation
whr_interest.describe()
```

Out[74]:

	year	life ladder	log gdp	social support	life expectancy	freedom	genero
count	2199.000000	2199.000000	2179.000000	2186.000000	2145.000000	2166.000000	2126.000
mean	2014.161437	5.479227	9.389760	0.810681	63.294582	0.747847	0.000
std	4.718736	1.125527	1.153402	0.120953	6.901104	0.140137	0.161
min	2005.000000	1.281000	5.527000	0.228000	6.720000	0.258000	-0.338
25%	2010.000000	4.647000	8.500000	0.747000	59.120000	0.656250	-0.112
50%	2014.000000	5.432000	9.499000	0.836000	65.050000	0.770000	-0.023
75%	2018.000000	6.309500	10.373500	0.905000	68.500000	0.859000	0.092
max	2022.000000	8.019000	11.664000	0.987000	74.475000	0.985000	0.703

Comparing that to dropping the na's altogether and we see that there is less difference in the data when imputing with iterpolation vs dropping the missing values altogether. We choose iterpolation and will set it to be our cleaned data that is ready to be used for regression.

In [75]:

```
# Choose interpolation and set to whr_clean
whr_clean = whr_interpolate

# Test dropping
whr_dropna = whr_interest.dropna()
whr_dropna.describe()
```

Out[75]:

	year	life ladder	log gdp	social support	life expectancy	freedom	genero
count	1964.000000	1964.000000	1964.000000	1964.000000	1964.000000	1964.000000	1964.000
mean	2014.289715	5.457233	9.342124	0.808017	63.153826	0.745511	0.000
std	4.647983	1.139948	1.157244	0.123474	7.059481	0.140401	0.161
min	2005.000000	2.179000	5.527000	0.290000	6.720000	0.258000	-0.338
25%	2010.750000	4.609000	8.440750	0.739000	58.593750	0.652750	-0.109
50%	2014.000000	5.391000	9.492000	0.834000	64.990000	0.767000	-0.023
75%	2018.000000	6.272250	10.297000	0.906000	68.560000	0.857000	0.090
max	2022.000000	7.971000	11.664000	0.987000	74.475000	0.985000	0.703

Visualize data

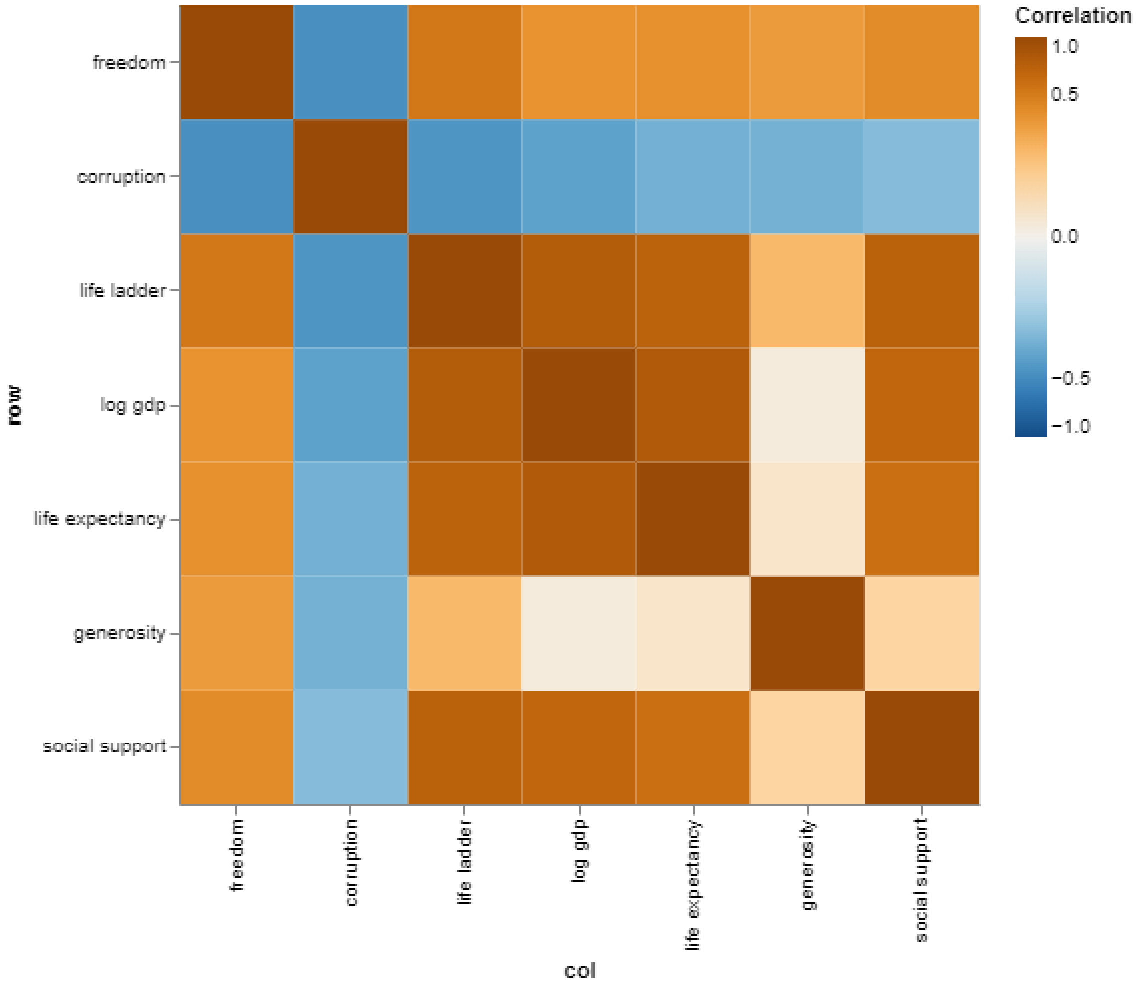
Let's begin by analyzing a correlation plot to see how correlated our explanatory variables are with eachother. This will be a heatmap.

In [76]:

```
corr_whr = whr_clean.drop(columns = ['year', 'country', 'continent'], axis = 0).
corr_whr.head()

#corr_mx.reset_index().columns
```

```
# melt corr_mx
corr_whr_long = corr_whr.reset_index().rename(
    columns = {'index':'row'}
    ).melt(
    id_vars = 'row',
    var_name = 'col',
    value_name = 'Correlation'
)

# construct heatmap
fig_corr_whr = alt.Chart(corr_whr_long).mark_rect().encode(
    x=alt.X('col', sort={'field': 'Correlation', 'order': 'ascending'}),
    y=alt.Y('row', sort={'field': 'Correlation', 'order': 'ascending'}),
    color=alt.Color('Correlation',
                    scale=alt.Scale(scheme='blueorange',
                                    domain=(-1, 1),
                                    type='sqrt'),
                    legend=alt.Legend(tickCount=5)
                    )
).properties(
    width=400,  # Adjust the width of the chart
    height=400  # Adjust the height of the chart
)

# display
fig_corr_whr
```

Out[76]:



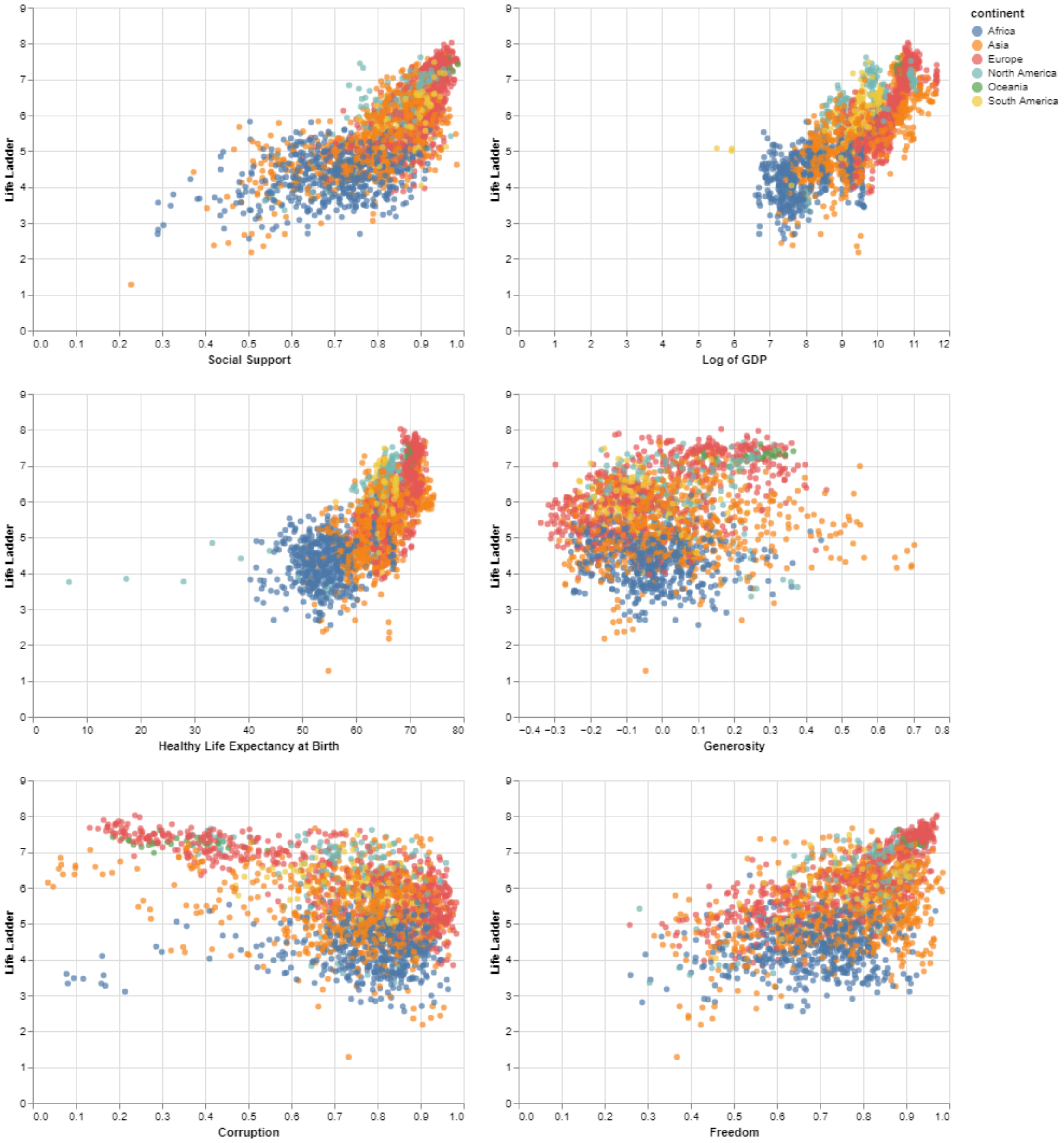
Interesting! Our correlation plot of the variables we are looking at in our analysis are showing some neat patterns. First and foremost, our metric for corruption is strongly negatively correlated with every other variable in our data set for world happiness. Corruption has the strongest negative correlations with freedom and the life ladder variable. Naturally, this would make sense in that corruption is almost exclusively a negative trait no matter the context in which it appears, so it is not surprising that it appears to have a detrimental effect on almost all other metrics widely considered to be at least somewhat related to happiness. Corruption leads to distrust among citizens of a nation, enables freedoms to be surpressed and often inhibits people from moving up socio-economically, as we can see with the negative correlation it shares with our log(GDP) variable.

The generosity variable is the variable which shares the least amount of relationship with the most of the other variables, as it is just mildly positively correlated with social support, life expectancy, log(GDP), life ladder, and freedom.

Freedom, life ladder, log(GDP), life expectancy, and social support all have positive correlations with one another, so as one metric increases, we may expect the other aforementioned variables to also trend in the same directions.

In [77]:
```
# Scatter plot of life ladder and social support
social_ladder = alt.Chart(whr_interest).mark_circle().encode(
    x=alt.X('social support', title = "Social Support"),
    y=alt.Y('life ladder',title = 'Life Ladder'),
    color = 'continent'
)

# Scatter plot of life ladder and log GDP per capita
gdp_ladder = alt.Chart(whr_interest).mark_circle().encode(
    x=alt.X('log gdp', title = 'Log of GDP'),
    y=alt.Y('life ladder',title = 'Life Ladder'),
    color = 'continent'
)

# Scatter plot of life ladder and life expectancy

expectancy_ladder = alt.Chart(whr_interest).mark_circle().encode(
    x = alt.X('life expectancy',
             title = 'Healthy Life Expectancy at Birth'),
    y = alt.Y('life ladder',
             title = 'Life Ladder'),
    color = 'continent',
)# , scale = alt.Scale(domain = [40,80])
#whr_clean
fig_gen_gdp = alt.Chart(whr_clean).mark_circle().encode(
    x= alt.X('generosity', title = "Generosity"),
    y = alt.Y('life ladder', title = 'Life Ladder'),
    color = 'continent'
)
fig_gen_gdp

fig_fre_lad = alt.Chart(whr_clean).mark_circle().encode(
    x= alt.X('freedom', title = "Freedom"),
    y = alt.Y('life ladder', title = 'Life Ladder'),
    color = 'continent'
)
fig_cor_lad = alt.Chart(whr_clean).mark_circle().encode(
    x= alt.X('corruption', title = "Corruption"),
    y = alt.Y('life ladder', title = 'Life Ladder'),
    color = 'continent'
)
# Combine and display the plots
h_fig1 = alt.hconcat(social_ladder, gdp_ladder)
h_fig2 = alt.hconcat(expectancy_ladder, fig_gen_gdp)
h_fig3 = alt.hconcat(fig_cor_lad, fig_fre_lad)

alt.vconcat(h_fig1, h_fig2, h_fig3)
```

Out[77]:



Interpretting Our Graphs

Above we have gone and taken a look at the relationship of our variable of interest, Life Ladder, against all of our predictor variables: Social Support, Log of GDP, Healthy Life Expectancy at Birth, Generosity, Corruption, and Freedom. Let's walk through our visualizations individually to do some deeper digging into the underlying patterns we can see.

Social Support and Life Ladder

In our graph of Social Support and Life Ladder, we can clearly see that for all non-African continents there is a dramatic increase in Life Ladder (ie. happiness) as the Social Support increases. As for the nations in Africa, it looks that such a trend is less significant and that social support actually has not the greatest effect in the Life Ladder of such nations. Europe, Asia, North America, and South America are the groups that exhibit the most clear positive relationship bwetween Social Support and Life Ladder.

Log of GDP and Life Ladder

The plot we have made of the Log of GDP and Life Ladder has an extremely visible trend. As goes the old adage, 'Money Buys Happiness'. Every single continent we have shows that as the GDP increases, as does the Life Ladder metric, and drastically.

Life Expectancy and Life Ladder

In comparing the Life Expectancy of our nations by continent against the Life Ladder, we can see there is a grouping of Life Expectancy by continent, with Africa being noticeably the lowest life expectancy. Depsite this, there is again a clearly positive trend in Life Ladder as the Healthy Life Expectancy at Birth increases.

Generosity and Life Ladder

Generosity has a less clear relationship with Life Ladder than most of our other variables, which as we had discussed above in the correlation plot. There is, however some interesting patterns, as regardless of Life Ladder, there is a large clustering of nations across all continents who sit on the negative end of our generosity metric. This tells me that despite varyong levels of happiness, people appreciate private ownership as is common in a globalized society as today. Conclusion, there is not much of a relationship between Generosity and Life Ladder.

Corruption and Life Ladder

As the correlation plot we had seen above describes, there is a strong negative relatinship between the levels of corruption a country faces with its Life Ladder metric of happiness. It is interesting to note that as the Life Ladder increases, regardless of the downwards trend, there is some grouping among continents that is easily visible. For example, the European and North American countries with higher Corruption indices are on average higher in the Life Ladder metric than nations in Africa and Asia.

Freedom and Life Ladder

At first glance, it is quick to note that the relationship between Freedom and Life Ladder is a positive one, as Freedom increases, Life Ladder does as well. African countries seem to exhibit less Freedom than most other nations and actually display the least dramatic increase in Life Ladder by Freedom. All other continents, on the other hand, do in fact seem to demonstrate the clear positive trend we have been discussing.

Fit a regression model

We will initially build a model that uses every variable of interest then use backward stepwise selection along with AIC to determine what our best model is.

In [78]:

```
# list our explanatory variables
X = whr_clean[['log gdp','social support', 'life expectancy', 'freedom', 'generc
# add an intercept term
X = sm.add_constant(X)
# assign the response variable
y = whr_clean['life ladder']

# fit the model
model_full = sm.OLS(y, X)
results_full = model_full.fit()
results_full.summary() # This shows how our data performed. Essetially extractin
```

Out[78]:

```
                         OLS Regression Results
     Dep. Variable:         life ladder          R-squared:    0.743
            Model:                  OLS     Adj. R-squared:    0.743
           Method:        Least Squares        F-statistic:    1059.
             Date:     Thu, 15 Jun 2023  Prob (F-statistic):    0.00
             Time:             04:50:24     Log-Likelihood:  -1884.1
 No. Observations:                 2199                AIC:    3782.
     Df Residuals:                 2192                BIC:    3822.
         Df Model:                    6
  Covariance Type:            nonrobust

                     coef  std err        t  P>|t|  [0.025  0.975]
          const   -2.0991    0.160  -13.153  0.000  -2.412  -1.786
        log gdp    0.3387    0.020   16.840  0.000   0.299   0.378
 social support    2.7995    0.142   19.759  0.000   2.522   3.077
life expectancy    0.0276    0.003    9.059  0.000   0.022   0.034
        freedom    1.1571    0.109   10.571  0.000   0.942   1.372
     generosity    0.5822    0.083    7.045  0.000   0.420   0.744
     corruption   -0.6511    0.079   -8.203  0.000  -0.807  -0.495

      Omnibus:  28.477    Durbin-Watson:     0.585
Prob(Omnibus):   0.000  Jarque-Bera (JB):   39.069
         Skew:  -0.158          Prob(JB):  3.28e-09
     Kurtosis:   3.571          Cond. No.      947.


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

After fitting the model and checking the summary of results we see that the full model has an AIC of 3782. The R^2 shows the goodness of fit for a regression model. In this case we have an $R^2 = 0.743$. Lets try a reduced model and see what changes.

```
In [79]: # Remove an explanatory variable
         # list our explanatory variables w/o corruption
         X = whr_clean[['log gdp','social support', 'life expectancy', 'freedom', 'generc

         # add an intercept term
         X = sm.add_constant(X)
         # fit the model
         model2 = sm.OLS(y, X)
         results2 = model2.fit()
         results2.summary()
```

```
Out[79]:                  OLS Regression Results
         Dep. Variable:        life ladder          R-squared:   0.736
                 Model:                OLS     Adj. R-squared:   0.735
                Method:      Least Squares        F-statistic:   1220.
                  Date:   Thu, 15 Jun 2023  Prob (F-statistic):  0.00
                  Time:           04:50:24     Log-Likelihood:  -1917.3
      No. Observations:               2199                AIC:   3847.
          Df Residuals:               2193                BIC:   3881.
              Df Model:                  5
       Covariance Type:          nonrobust

                           coef  std err        t  P>|t|  [0.025  0.975]
                  const -2.9879    0.119  -25.122  0.000  -3.221  -2.755
                log gdp  0.3809    0.020   19.304  0.000   0.342   0.420
         social support  2.6428    0.142   18.547  0.000   2.363   2.922
        life expectancy  0.0260    0.003    8.424  0.000   0.020   0.032
                freedom  1.4752    0.104   14.200  0.000   1.271   1.679
             generosity  0.7105    0.082    8.626  0.000   0.549   0.872

              Omnibus: 21.575    Durbin-Watson:    0.572
        Prob(Omnibus):  0.000  Jarque-Bera (JB):  28.531
                 Skew: -0.130          Prob(JB):  6.38e-07
             Kurtosis:  3.493          Cond. No.    761.
```

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

Removing a single explanatory variable actually increased our AIC and decreased our R^2. After removing the variable `corruption`, our AIC jumped up to 3847 and our R^2 decreased to 0.736. This is a sign that we should select the full model as our best fit. I would assume that this is due to the high amount of correlation between the variables in the data. Let's go back to the summary output of our full model and analyze the findings.

```
In [80]: results_full.summary()
```

```
Out[80]:
                          OLS Regression Results
       Dep. Variable:        life ladder          R-squared:     0.743
              Model:                 OLS     Adj. R-squared:     0.743
             Method:       Least Squares        F-statistic:     1059.
               Date:    Thu, 15 Jun 2023  Prob (F-statistic):     0.00
               Time:            04:50:24     Log-Likelihood:   -1884.1
  No. Observations:                 2199                AIC:     3782.
       Df Residuals:                2192                BIC:     3822.
           Df Model:                   6
   Covariance Type:            nonrobust

                       coef   std err         t   P>|t|   [0.025   0.975]
           const    -2.0991     0.160   -13.153   0.000   -2.412   -1.786
         log gdp     0.3387     0.020    16.840   0.000    0.299    0.378
  social support     2.7995     0.142    19.759   0.000    2.522    3.077
 life expectancy     0.0276     0.003     9.059   0.000    0.022    0.034
         freedom     1.1571     0.109    10.571   0.000    0.942    1.372
      generosity     0.5822     0.083     7.045   0.000    0.420    0.744
      corruption    -0.6511     0.079    -8.203   0.000   -0.807   -0.495

        Omnibus:  28.477    Durbin-Watson:      0.585
  Prob(Omnibus):   0.000  Jarque-Bera (JB):    39.069
           Skew:  -0.158         Prob(JB):  3.28e-09
       Kurtosis:   3.571         Cond. No.      947.

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

The results from the regression analysis indicate that the model has a strong overall fit, with an R-squared value of 0.743. This means that approximately 74.3% of the variation in the dependent variable, life ladder, can be explained by the independent variables included in the model. The adjusted R-squared value, which accounts for the number of predictors and sample size, is also 0.743. The F-statistic of 1059 suggests that the overall model is statistically significant, with a p-value of 0.00.

Examining the coefficients of the individual predictors, it can be observed that each variable has a significant impact on the life ladder outcome. The constant term (intercept) is -2.0991, indicating that when all predictors are zero, the expected value of life ladder is -2.0991. The log gdp, social support, life expectancy, freedom, generosity, and corruption predictors have coefficients of 0.3387, 2.7995, 0.0276, 1.1571, 0.5822, and -0.6511, respectively. These coefficients represent the expected change in the life ladder for a one-unit increase in the corresponding predictor, holding other predictors constant.

The standard errors, t-values, and p-values associated with each coefficient indicate the precision and statistical significance of the estimates. All predictors have p-values of 0.000, suggesting that they are statistically significant in explaining the life ladder outcome. The 95% confidence intervals, given by the [0.025, 0.975] range, do not include zero for any predictor, further supporting their significance.

To efficiently estimate life ladder across all countries our model uses the following equation:

$$\begin{gathered}\text{Life Ladder} = -2.0991 + 0.3387 * \text{log gdp} + 2.7995 * \text{social support} \\ + 0.0276 * \text{life expectancy} + 1.1571 * \text{freedom} + 0.5822 * \text{generosity} \\ - 0.6511 * \text{corruption}\end{gathered}$$

Summary of findings

Lets start by re-assessing our question of interest: "With what subset of metrics of the quality of life in nations around the world- GDP, freedom, social support, corruption, life expectancy, etc.- can we best predict the nation's happiness level as measured by our variable called Life Ladder. How does each quality of life metric affect life ladder nations in each continent?" We also stated "An ideal answer to this question would be to show the overall trend in world happiness, then be broken down to continents happiness levels. We would then estimate life ladder based off the best combination of expanatory variables that are chosen with model selection techniques." In our Data Analysis, we answered these questions using visualzation and Ordinary Least Squares to estimate the overall trend.

We found that:

- There is a heavy correlation between all explanatory variable.
- Generosity has little to no relationship with life ladder.
- Corruption has a visibly negative relationship with life ladder.
- Freedom has a somewhat positive relationship with life ladder.
- Social support, log(GDP), Healthy life expectancy at birth have extremely positive relationships with life ladder.
- North America and Europe tend to rank the highest in life ladder.
- Africa has a ceiling at about 6 on the life ladder metric across all graphs.

- Africa seems to exhibit the least trend out of all continents regardless of which qualtiy of life metric is used.
- A regression model consiting of all 6 independent variables is the most effective in estimating the global life ladder metric.
- The best model uses this equation:

$$\begin{gathered}\text{Life Ladder} = -2.0991 + 0.3387 * \text{log gdp} + 2.7995 * \text{social support} \\ + 0.0276 * \text{life expectancy} + 1.1571 * \text{freedom} + 0.5822 * \text{generosity} \\ - 0.6511 * \text{corruption}\end{gathered}$$