30538 Final Project

Sohyun Lim and Ting Tsai

2024-12-05

1. Load the dataset

```
import pandas as pd

# 1. Load the dataset.
file_path = "/Users/sohyunlim/Desktop/python -
↪  final/Public_Health_Statistics_-_Selected_public_health_indicators_by_Chicago_community_are
data = pd.read_csv(file_path)
data.head()


# 2. Change the variable names.
# Replace parentheses with empty strings, hyphens with underscores, and spaces
↪  with underscores in column names
data.columns = data.columns.str.replace(r"[()]", "", regex=True).str.replace("-",
↪  "_").str.replace(" ", "_").str.lower()
print(data.columns)


# 3. Remove unncessary columns.
columns_to_drop = [
    "childhood_blood_lead_level_screening",
    "childhood_lead_poisoning",
    "gonorrhea_in_females"
]
data = data.drop(columns=columns_to_drop)
print(data.columns)
```

```
Index(['community_area', 'community_area_name', 'birth_rate',
       'general_fertility_rate', 'low_birth_weight',
       'prenatal_care_beginning_in_first_trimester', 'preterm_births',
       'teen_birth_rate', 'assault_homicide', 'breast_cancer_in_females',
       'cancer_all_sites', 'colorectal_cancer', 'diabetes_related',
       'firearm_related', 'infant_mortality_rate', 'lung_cancer',
```

```
       'prostate_cancer_in_males', 'stroke_cerebrovascular_disease',
       'childhood_blood_lead_level_screening', 'childhood_lead_poisoning',
       'gonorrhea_in_females', 'gonorrhea_in_males', 'tuberculosis',
       'below_poverty_level', 'crowded_housing', 'dependency',
       'no_high_school_diploma', 'per_capita_income', 'unemployment'],
      dtype='object')
Index(['community_area', 'community_area_name', 'birth_rate',
       'general_fertility_rate', 'low_birth_weight',
       'prenatal_care_beginning_in_first_trimester', 'preterm_births',
       'teen_birth_rate', 'assault_homicide', 'breast_cancer_in_females',
       'cancer_all_sites', 'colorectal_cancer', 'diabetes_related',
       'firearm_related', 'infant_mortality_rate', 'lung_cancer',
       'prostate_cancer_in_males', 'stroke_cerebrovascular_disease',
       'gonorrhea_in_males', 'tuberculosis', 'below_poverty_level',
       'crowded_housing', 'dependency', 'no_high_school_diploma',
       'per_capita_income', 'unemployment'],
      dtype='object')
```

3. Basic Plots (table and graph)

To explore the community area by health outcomes and socioeconomic factors

3-1. Create a table of cancer_all_sites (descending) by community area

3-2. Create a table of diabetes_related (descending) by community area

3-3. Create a table of tuberculosis (descending) by community area

3-4. Create a table of below_poverty_level (descending) by community area

3-5. Create a table of no_high_school_diploma (descending) by community area

3-6. Create a table of per_capita_income (descending) by community area

3-7. Create a table of unemployment (descending) by community area

```
# 3-1. Create a table of cancer_all_sites (descending) by community area
import pandas as pd
import altair as alt

# Select necessary columns from dataframe
cancer_table = data[["community_area", "community_area_name",
 ↪  "cancer_all_sites"]].copy()

# Drop na and arrange by descending
cancer_table = cancer_table.dropna(subset=["cancer_all_sites"])
cancer_table = cancer_table.sort_values(by="cancer_all_sites", ascending=False)

# Initiate index
cancer_table.reset_index(drop=True, inplace=True)
```

```
# Print the result
print(cancer_table)


# 3-2. Create a table of diabetes_related (descending) by community area
# Select necessary columns from dataframe
diabetes_table = data[["community_area", "community_area_name",
 ↪  "diabetes_related"]].copy()

# Drop na and arrange by descending
diabetes_table = diabetes_table.dropna(subset=["diabetes_related"])
diabetes_table = diabetes_table.sort_values(by="diabetes_related",
 ↪  ascending=False)

# Initiate index
diabetes_table.reset_index(drop=True, inplace=True)

# Print the result
print(diabetes_table)

# 3-3. Create a table of tuberculosis (descending) by community area
# Select necessary columns from dataframe
tuberculosis_table = data[["community_area", "community_area_name",
 ↪  "tuberculosis"]].copy()

# Drop na and arrange by descending
tuberculosis_table = tuberculosis_table.dropna(subset=["tuberculosis"])
tuberculosis_table = tuberculosis_table.sort_values(by="tuberculosis",
 ↪  ascending=False)

# Initiate index
tuberculosis_table.reset_index(drop=True, inplace=True)

# Print the result
print(tuberculosis_table)


# 3-4. Create a table of below_poverty_level (descending) by community area
# Select necessary columns from dataframe
poverty_table = data[["community_area", "community_area_name",
 ↪  "below_poverty_level"]].copy()

# Drop na and arrange by descending
poverty_table = poverty_table.dropna(subset=["below_poverty_level"])
poverty_table = poverty_table.sort_values(by="below_poverty_level",
 ↪  ascending=False)
```

```python
# Initiate index
poverty_table.reset_index(drop=True, inplace=True)

# Print the result
print(poverty_table)


# 3-5. Create a table of no_high_school_diploma (descending) by community area
# Select necessary columns from dataframe
education_table = data[["community_area", "community_area_name",
 ↪  "no_high_school_diploma"]].copy()

# Drop na and arrange by descending
education_table = education_table.dropna(subset=["no_high_school_diploma"])
education_table = education_table.sort_values(by="no_high_school_diploma",
 ↪  ascending=False)

# Initiate index
education_table.reset_index(drop=True, inplace=True)

# Print the result
print(education_table)


# 3-6. Create a table of per_capita_income (descending) by community area
# Select necessary columns from dataframe
income_table = data[["community_area", "community_area_name",
 ↪  "per_capita_income"]].copy()

# Drop na and arrange by descending
income_table = income_table.dropna(subset=["per_capita_income"])
income_table = income_table.sort_values(by="per_capita_income", ascending=False)

# Initiate index
income_table.reset_index(drop=True, inplace=True)

# Print the result
print(income_table)


# 3-7. Create a table of unemployment (descending) by community area
# Select necessary columns from dataframe
unemployment_table = data[["community_area", "community_area_name",
 ↪  "unemployment"]].copy()

# Drop na and arrange by descending
unemployment_table = unemployment_table.dropna(subset=["unemployment"])
```

```python
unemployment_table = unemployment_table.sort_values(by="unemployment",
    ascending=False)

# Initiate index
unemployment_table.reset_index(drop=True, inplace=True)

# Print the result
print(unemployment_table)
```

```
    community_area     community_area_name  cancer_all_sites
0               26      West Garfield Park             291.5
1               69  Greater Grand Crossing             274.4
2               35                 Douglas             269.9
3               53            West Pullman             263.6
4               50                 Pullman             262.5
..             ...                     ...               ...
72              20                 Hermosa             135.2
73              21                Avondale             133.9
74              30           South Lawndale            127.4
75               6               Lake View             126.9
76              32                    Loop             120.1

[77 rows x 3 columns]
    community_area community_area_name  diabetes_related
0               35             Douglas             119.1
1               26  West Garfield Park             118.2
2               54           Riverdale             115.9
3               25              Austin             113.9
4               37         Fuller Park             111.7
..             ...                 ...               ...
72               6           Lake View              38.5
73              12         Forest Glen              37.2
74              41           Hyde Park              34.0
75               8     Near North Side              27.0
76              32                Loop              26.8

[77 rows x 3 columns]
    community_area community_area_name  tuberculosis
0               34       Armour Square          22.7
1               42            Woodlawn          17.4
2               14         Albany Park          16.8
3               20             Hermosa          15.8
4               28      Near West Side          14.1
..             ...                 ...           ...
72              64            Clearing           0.9
73              72             Beverly           0.0
74              74      Mount Greenwood          0.0
```

```
75             37          Fuller Park          0.0
76             39              Kenwood          0.0

[77 rows x 3 columns]
    community_area community_area_name  below_poverty_level
0               54           Riverdale                 61.4
1               37         Fuller Park                 55.5
2               68           Englewood                 42.2
3               26   West Garfield Park                 40.3
4               27   East Garfield Park                 39.7
..             ...                 ...                  ...
72              64            Clearing                  5.9
73              10         Norwood Park                  5.9
74              72             Beverly                  5.2
75               9         Edison Park                  5.1
76              74      Mount Greenwood                  3.1

[77 rows x 3 columns]
    community_area community_area_name  no_high_school_diploma
0               30       South Lawndale                    58.7
1               63            Gage Park                    54.1
2               58        Brighton Park                    48.2
3               31      Lower West Side                    44.3
4               61             New City                    42.4
..             ...                 ...                     ...
72              74      Mount Greenwood                     4.5
73               7         Lincoln Park                     4.3
74              32                 Loop                     3.4
75               8      Near North Side                     3.4
76               6            Lake View                     2.9

[77 rows x 3 columns]
    community_area community_area_name  per_capita_income
0                8      Near North Side              87163
1                7         Lincoln Park              71403
2               32                 Loop              67699
3               33      Near South Side              60593
4                6            Lake View              58227
..             ...                 ...                ...
72              26   West Garfield Park              10951
73              30       South Lawndale              10697
74              67       West Englewood              10559
75              37          Fuller Park               9016
76              54            Riverdale               8535

[77 rows x 3 columns]
    community_area community_area_name  unemployment
0               37          Fuller Park          40.0
```

```
1                 67      West Englewood          34.7
2                 36             Oakland          26.6
3                 54           Riverdale          26.4
4                 26   West Garfield Park         25.2
..               ...                 ...           ...
72                 6           Lake View           4.7
73                76              O'Hare           4.7
74                 7        Lincoln Park           4.5
75                 5        North Center           4.5
76                32                Loop           4.2

[77 rows x 3 columns]
```

Select top 10 areas in each table and converty it into a graph.

3-8. Create a table of cancer_all_sites (descending) by community area

3-9. Create a table of diabetes_related (descending) by community area

3-10. Create a table of tuberculosis (descending) by community area

3-11. Create a table of below_poverty_level (descending) by community area

3-12. Create a table of no_high_school_diploma (descending) by community area

3-13. Create a table of per_capita_income (descending) by community area

3-14. Create a table of unemployment (descending) by community area

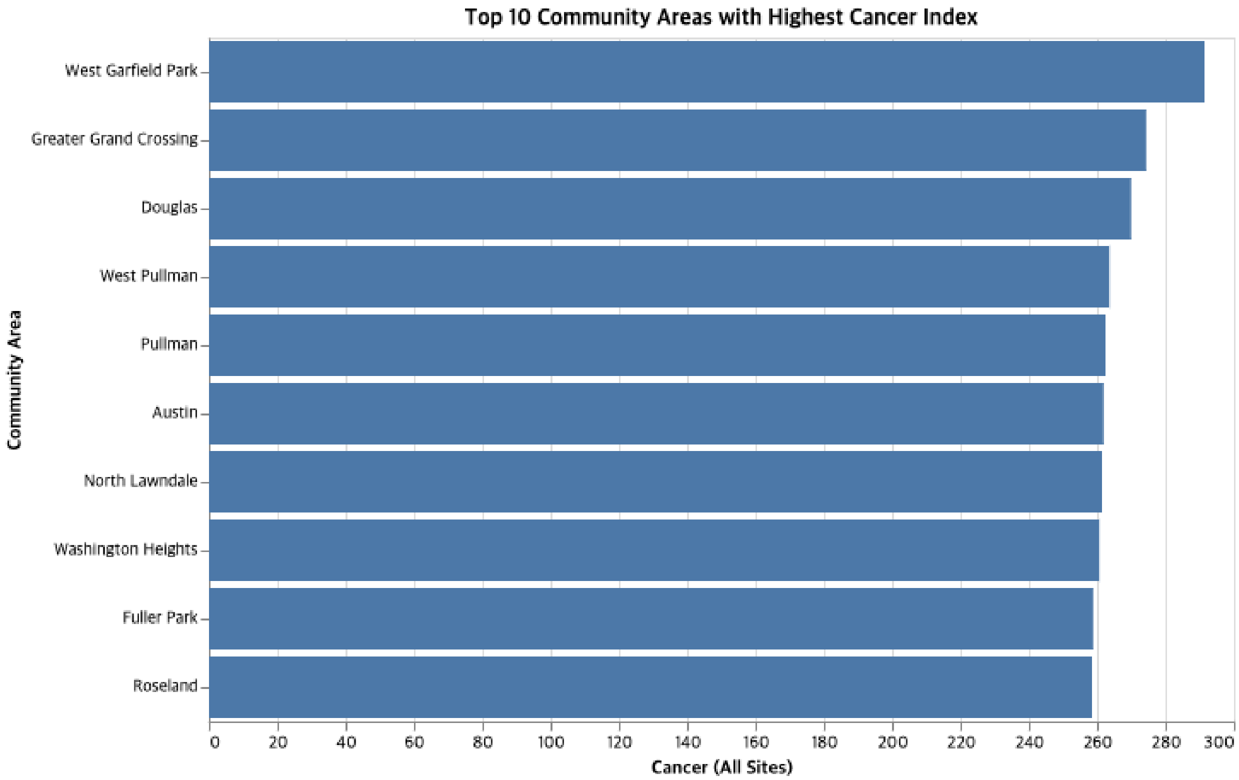
```
# 3-8. Create a table of cancer_all_sites (descending) by community area
# Extractn top 10 community areas
top_10_cancer_areas = cancer_table.head(10)

# Create a bar chart
```

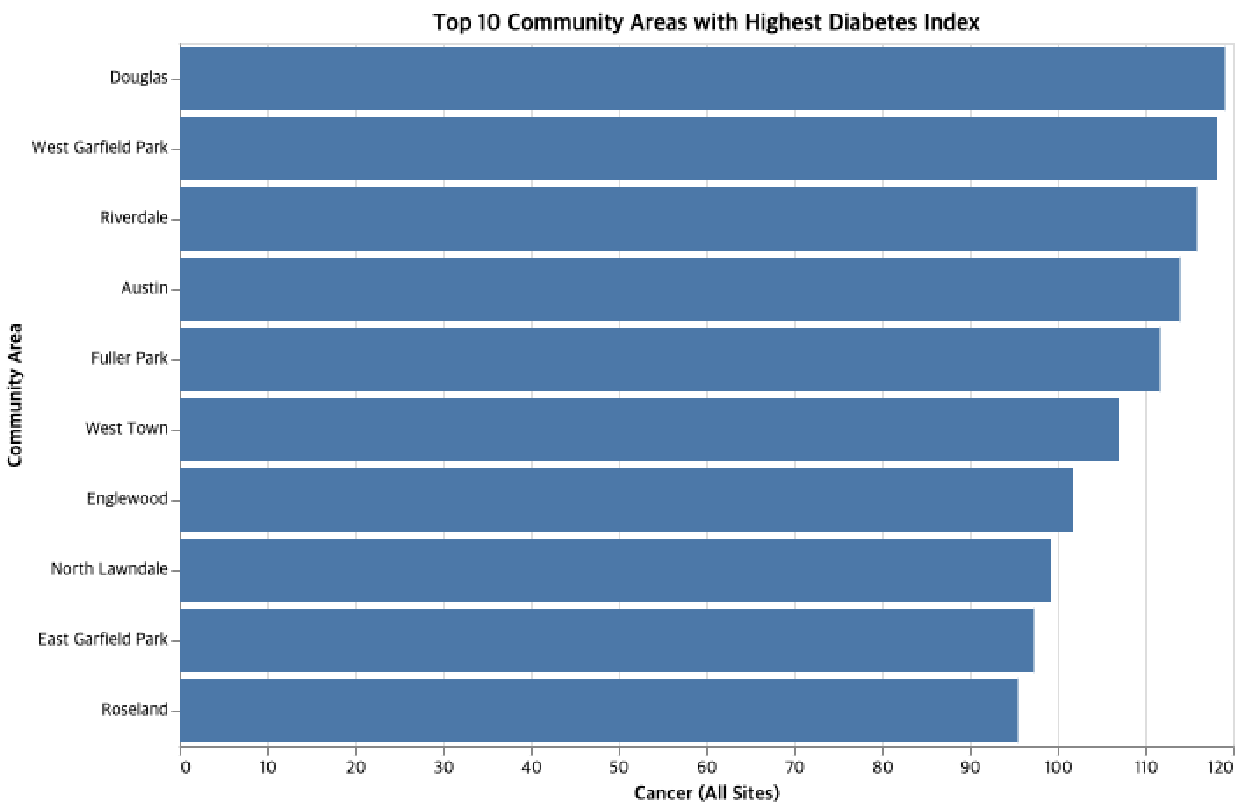
```
bar_chart_cancer = alt.Chart(top_10_cancer_areas).mark_bar().encode(
    x=alt.X("cancer_all_sites:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "cancer_all_sites"]
).properties(
    title="Top 10 Community Areas with Highest Cancer Index",
    width=600,
    height=400
)

bar_chart_cancer.show()


# 3-9. Create a table of diabetes_related (descending) by community area
# Extractn top 10 community areas
top_10_diabetes_areas = diabetes_table.head(10)

# Create a bar chart
bar_chart_diabetes = alt.Chart(top_10_diabetes_areas).mark_bar().encode(
    x=alt.X("diabetes_related:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "diabetes_related"]
).properties(
    title="Top 10 Community Areas with Highest Diabetes Index",
    width=600,
    height=400
)

bar_chart_diabetes.show()


# 3-10. Create a table of tuberculosis (descending) by community area
# Extractn top 10 community areas
top_10_tuberculosis_areas = tuberculosis_table.head(10)

# Create a bar chart
bar_chart_tuberculosis = alt.Chart(top_10_tuberculosis_areas).mark_bar().encode(
    x=alt.X("tuberculosis:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "tuberculosis"]
).properties(
    title="Top 10 Community Areas with Highest Tuberculosis Index",
    width=600,
    height=400
)

bar_chart_tuberculosis.show()
```

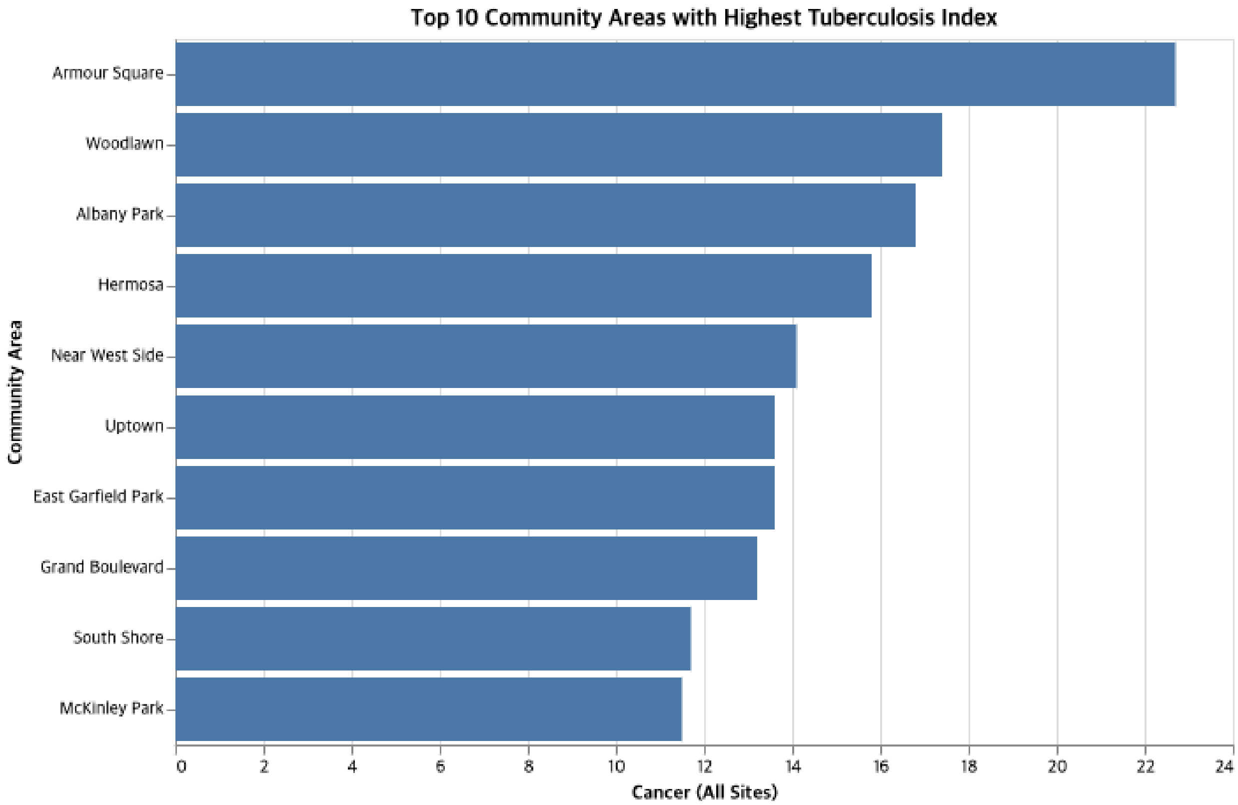
```
# 3-11. Create a table of below_poverty_level (descending) by community area
# Extractn top 10 community areas
top_10_poverty_areas = poverty_table.head(10)

# Create a bar chart
bar_chart_poverty = alt.Chart(top_10_poverty_areas).mark_bar().encode(
    x=alt.X("below_poverty_level:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "below_poverty_level"]
).properties(
    title="Top 10 Community Areas with Highest Poverty Index",
    width=600,
    height=400
)

bar_chart_poverty.show()


# 3-12. Create a table of no_high_school_diploma (descending) by community area
# Extractn top 10 community areas
top_10_education_areas = education_table.head(10)

# Create a bar chart
bar_chart_education = alt.Chart(top_10_education_areas).mark_bar().encode(
    x=alt.X("no_high_school_diploma:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "no_high_school_diploma"]
).properties(
    title="Top 10 Community Areas with Highest No Education Index",
    width=600,
    height=400
)

bar_chart_education.show()


# 3-13. Create a table of per_capita_income (descending) by community area
# Extractn top 10 community areas
top_10_income_areas = income_table.head(10)

# Create a bar chart
bar_chart_income = alt.Chart(top_10_income_areas).mark_bar().encode(
    x=alt.X("per_capita_income:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "per_capita_income"]
```

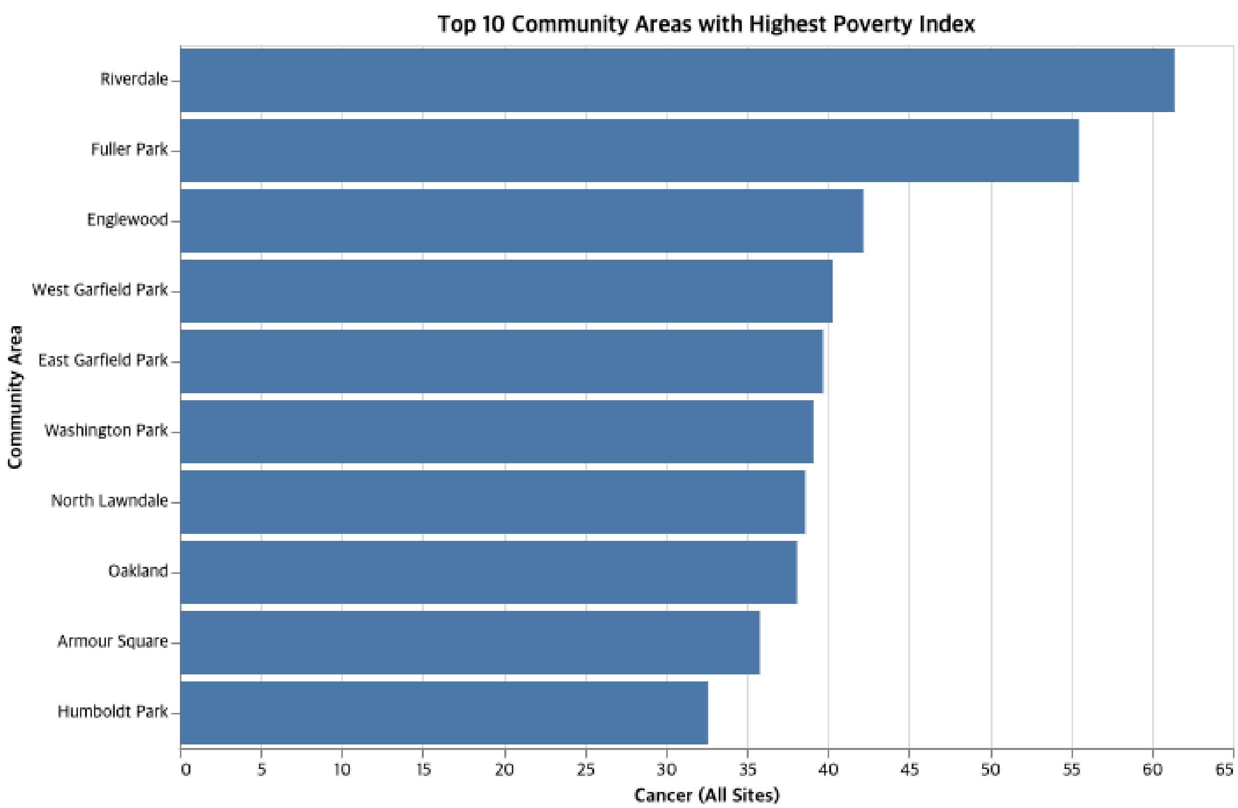
```python
).properties(
    title="Top 10 Community Areas with Highest Income (per capita) Index",
    width=600,
    height=400
)

bar_chart_income.show()


# 3-14. Create a table of unemployment (descending) by community area
# Extractn top 10 community areas
top_10_unemployment_areas = unemployment_table.head(10)

# Create a bar chart
bar_chart_unemployment = alt.Chart(top_10_unemployment_areas).mark_bar().encode(
    x=alt.X("unemployment:Q", title="Cancer (All Sites)"),
    y=alt.Y("community_area_name:N", sort="-x", title="Community Area"),
    tooltip=["community_area_name", "unemployment"]
).properties(
    title="Top 10 Community Areas with Highest Unemployment Index",
    width=600,
    height=400
)

bar_chart_unemployment.show()
```

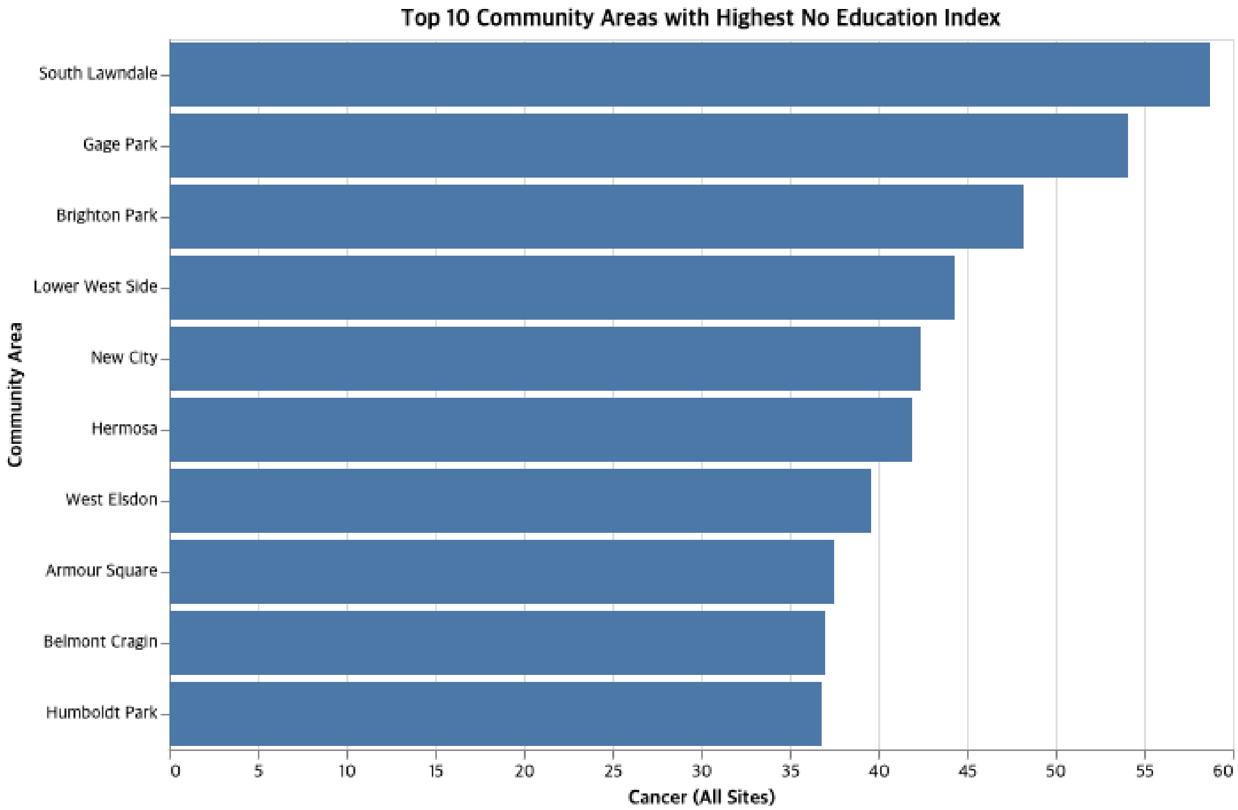
```
alt.Chart(...)

alt.Chart(...)

alt.Chart(...)

alt.Chart(...)

alt.Chart(...)

alt.Chart(...)

alt.Chart(...)
```

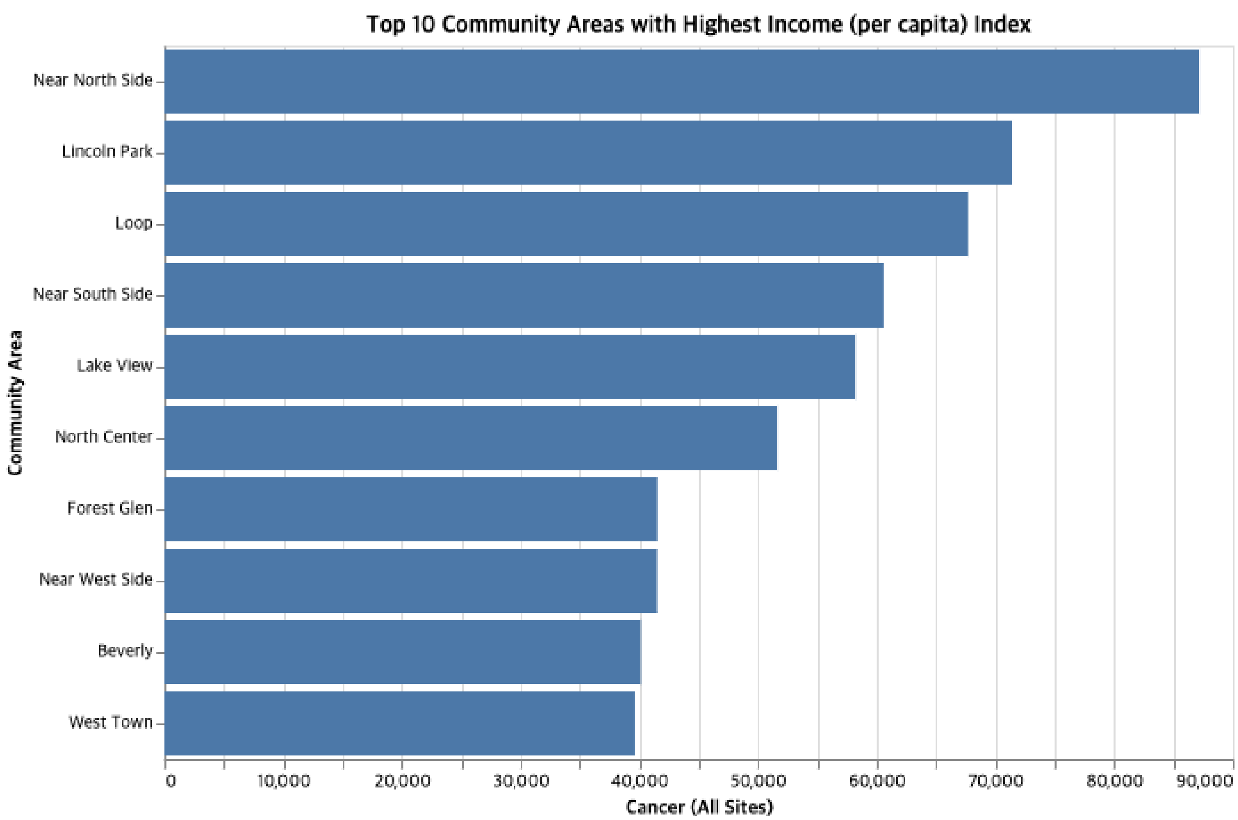

Top 10 Community Areas with Highest Cancer Index
Community Area
West Garfield Park
Greater Grand Crossing
Douglas
West Pullman
Pullman
Austin
North Lawndale
Washington Heights
Fuller Park
Roseland
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
Cancer (All Sites)

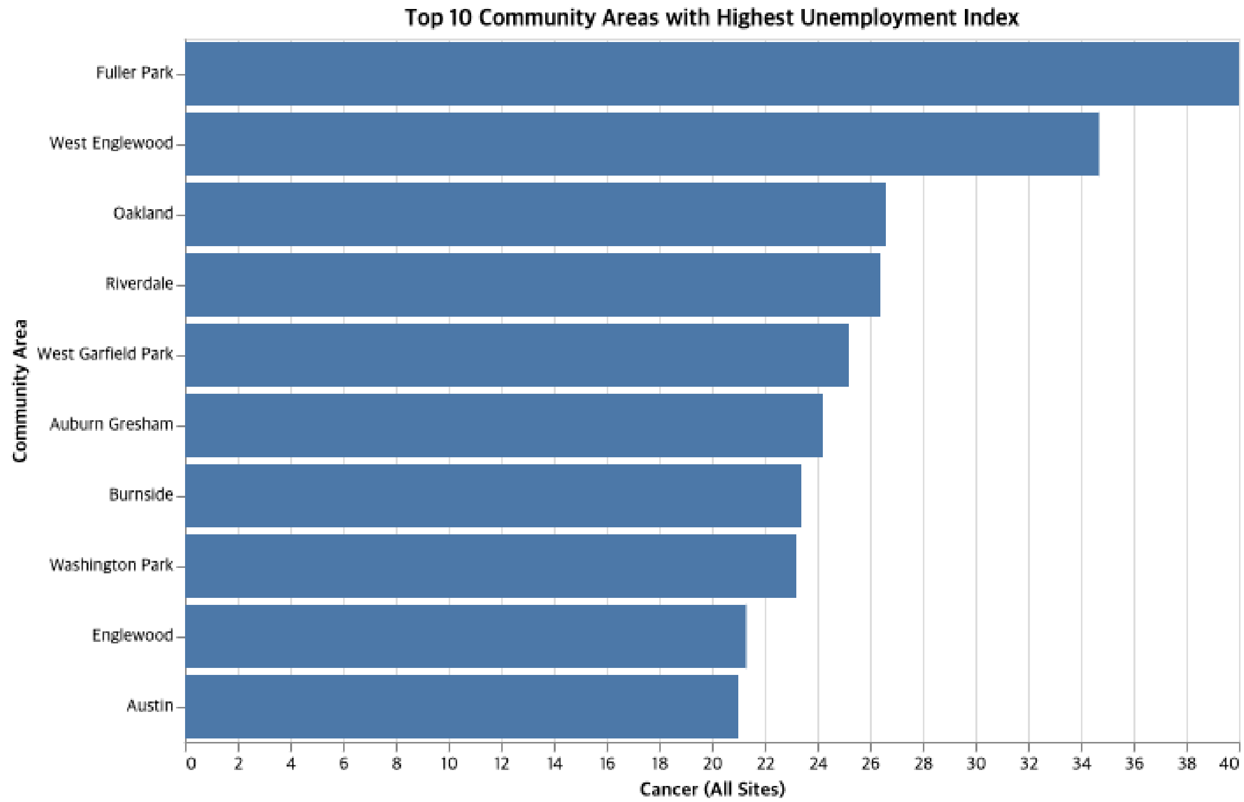

Top 10 Community Areas with Highest Diabetes Index
Community Area
Douglas
West Garfield Park
Riverdale
Austin
Fuller Park
West Town
Englewood
North Lawndale
East Garfield Park
Roseland
0
10
20
30
40
50
60
70
80
90
100
110
120
Cancer (All Sites)


Top 10 Community Areas with Highest Tuberculosis Index
Community Area
Armour Square
Woodlawn
Albany Park
Hermosa
Near West Side
Uptown
East Garfield Park
Grand Boulevard
South Shore
McKinley Park
0
2
4
6
8
10
12
14
16
18
20
22
24
Cancer (All Sites)


Top 10 Community Areas with Highest Poverty Index
Community Area
Riverdale
Fuller Park
Englewood
West Garfield Park
East Garfield Park
Washington Park
North Lawndale
Oakland
Armour Square
Humboldt Park
0
5
10
15
20
25
30
35
40
45
50
55
60
65
Cancer (All Sites)


Top 10 Community Areas with Highest No Education Index
Community Area
South Lawndale
Gage Park
Brighton Park
Lower West Side
New City
Hermosa
West Elsdon
Armour Square
Belmont Cragin
Humboldt Park
0
5
10
15
20
25
30
35
40
45
50
55
60
Cancer (All Sites)


Top 10 Community Areas with Highest Income (per capita) Index
Community Area
Near North Side
Lincoln Park
Loop
Near South Side
Lake View
North Center
Forest Glen
Near West Side
Beverly
West Town
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
Cancer (All Sites)



Show a table with some variables and community area.

(This can be used in a Shiny app: Select community area $>$ show table.)

```
# Select necessary columns from dataframe
variables_table = data[["community_area", "community_area_name",
↪   "cancer_all_sites", "diabetes_related", "tuberculosis",
↪   "below_poverty_level", "no_high_school_diploma", "per_capita_income",
↪   "unemployment"]].copy()

# Print the result
print(variables_table)
```

```
    community_area community_area_name  cancer_all_sites  diabetes_related  \
0                1         Rogers Park             176.9              77.1
1                2          West Ridge             155.9              60.5
2                3              Uptown             183.3              80.0
3                4      Lincoln Square             153.2              55.4
4                5        North Center             152.1              49.8
..             ...                 ...               ...               ...
72              73  Washington Heights             260.6              79.5
73              74      Mount Greenwood             201.1              66.5
```

```
74              75         Morgan Park            218.2              75.4
75              76              O'Hare            138.5              47.3
76              77           Edgewater            162.0              48.8

    tuberculosis  below_poverty_level  no_high_school_diploma  \
0           11.4                 22.7                    18.1
1            8.9                 15.1                    19.6
2           13.6                 22.7                    13.6
3            8.5                  9.5                    12.5
4            1.9                  7.1                     5.4
..           ...                  ...                     ...
72           3.0                 15.7                    15.6
73           0.0                  3.1                     4.5
74           2.6                 13.7                    10.9
75           6.3                  9.5                    11.0
76          10.5                 16.6                     9.0

    per_capita_income  unemployment
0               23714           7.5
1               21375           7.9
2               32355           7.7
3               35503           6.8
4               51615           4.5
..                ...           ...
72              19709          18.3
73              34221           6.9
74              26185          14.9
75              29402           4.7
76              33364           9.0

[77 rows x 9 columns]
```

2. Regression

2-2. Causal Relationship

Independent Variable (X) : below_poverty_level, no_high_school_diploma, per_capita_income, unemployment

Dependent Variable (Y) : cancer_all_sites, diabetes_related, tuberculosis, breast_cancer_in_females,colorectal_cancer, infant_mortality_rate, lung_cancer, prostate_cancer_in_males, stroke_cerebrovascular_disease,

```
import statsmodels.api as sm

# 1-1. below_poverty_level ~ cancer_all_sites
x = sm.add_constant(data["below_poverty_level"])
y = data["cancer_all_sites"]
model_1 = sm.OLS(y, x).fit()
print(model_1.summary())

# 1-2. no_high_school_diploma ~ cancer_all_sites
x = sm.add_constant(data["no_high_school_diploma"])
y = data["cancer_all_sites"]
model_2 = sm.OLS(y, x).fit()
print(model_2.summary())

# 1-3. per_capita_income ~ cancer_all_sites
x = sm.add_constant(data["per_capita_income"])
y = data["cancer_all_sites"]
model_3 = sm.OLS(y, x).fit()
print(model_3.summary())

# 1-4. unemployment ~ cancer_all_sites
x = sm.add_constant(data["unemployment"])
y = data["cancer_all_sites"]
model_4 = sm.OLS(y, x).fit()
print(model_4.summary())

# 2-1. below_poverty_level ~ diabetes_related
x = sm.add_constant(data["below_poverty_level"])
y = data["diabetes_related"]
model_5 = sm.OLS(y, x).fit()
print(model_5.summary())
```

```python
# 2-2. no_high_school_diploma ~ diabetes_related
x = sm.add_constant(data["no_high_school_diploma"])
y = data["diabetes_related"]
model_6 = sm.OLS(y, x).fit()
print(model_6.summary())

# 2-3. per_capita_income ~ diabetes_related
x = sm.add_constant(data["per_capita_income"])
y = data["diabetes_related"]
model_7 = sm.OLS(y, x).fit()
print(model_7.summary())

# 2-4. unemployment ~ diabetes_related
x = sm.add_constant(data["unemployment"])
y = data["diabetes_related"]
model_8 = sm.OLS(y, x).fit()
print(model_8.summary())

# 3-1. below_poverty_level ~ tuberculosis
x = sm.add_constant(data["below_poverty_level"])
y = data["tuberculosis"]
model_9 = sm.OLS(y, x).fit()
print(model_9.summary())

# 3-2. no_high_school_diploma ~ tuberculosis
x = sm.add_constant(data["no_high_school_diploma"])
y = data["tuberculosis"]
model_10 = sm.OLS(y, x).fit()
print(model_10.summary())

# 3-3. per_capita_income ~ tuberculosis
x = sm.add_constant(data["per_capita_income"])
y = data["tuberculosis"]
model_11 = sm.OLS(y, x).fit()
print(model_11.summary())

# 3-4. unemployment ~ tuberculosis
x = sm.add_constant(data["unemployment"])
y = data["tuberculosis"]
model_12 = sm.OLS(y, x).fit()
print(model_12.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       cancer_all_sites   R-squared:                       0.255
Model:                            OLS   Adj. R-squared:                  0.246
```

```
Method:                  Least Squares   F-statistic:                     25.73
Date:                 Mon, 02 Dec 2024   Prob (F-statistic):           2.75e-06
Time:                         20:59:09   Log-Likelihood:                -391.62
No. Observations:                   77   AIC:                             787.2
Df Residuals:                       75   BIC:                             791.9
Df Model:                            1
Covariance Type:            nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------------
const                153.5515      9.213     16.668      0.000     135.199
171.904
below_poverty_level    2.0070      0.396      5.073      0.000       1.219
2.795
==============================================================================
Omnibus:                        7.011   Durbin-Watson:                   1.337
Prob(Omnibus):                  0.030   Jarque-Bera (JB):                2.777
Skew:                           0.075   Prob(JB):                        0.249
Kurtosis:                       2.082   Cond. No.                         47.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:       cancer_all_sites   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.013
Method:                 Least Squares   F-statistic:                   0.01107
Date:                Mon, 02 Dec 2024   Prob (F-statistic):              0.916
Time:                        20:59:09   Log-Likelihood:                -402.97
No. Observations:                  77   AIC:                             809.9
Df Residuals:                      75   BIC:                             814.6
Df Model:                           1
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025
0.975]
-----------------------------------------------------------------------------------------
const                   195.2472     10.598     18.423      0.000     174.135
216.359
no_high_school_diploma   -0.0449      0.427     -0.105      0.916      -0.895
0.805
==============================================================================
Omnibus:                       21.703   Durbin-Watson:                   1.195
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                5.435
Skew:                           0.277   Prob(JB):                       0.0660
```

```
Kurtosis:                       1.823   Cond. No.                         50.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:       cancer_all_sites   R-squared:                       0.195
Model:                            OLS   Adj. R-squared:                  0.184
Method:                 Least Squares   F-statistic:                     18.13
Date:                Mon, 02 Dec 2024   Prob (F-statistic):           5.90e-05
Time:                        20:59:09   Log-Likelihood:                -394.64
No. Observations:                  77   AIC:                             793.3
Df Residuals:                      75   BIC:                             798.0
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025
0.975]
-------------------------------------------------------------------------------------
const              228.1009      9.229     24.715      0.000     209.715
246.486
per_capita_income   -0.0013      0.000     -4.258      0.000      -0.002
-0.001
==============================================================================
Omnibus:                       12.279   Durbin-Watson:                   1.217
Prob(Omnibus):                  0.002   Jarque-Bera (JB):                3.672
Skew:                          -0.098   Prob(JB):                        0.159
Kurtosis:                       1.948   Cond. No.                     5.73e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.73e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
Dep. Variable:       cancer_all_sites   R-squared:                       0.479
Model:                            OLS   Adj. R-squared:                  0.472
Method:                 Least Squares   F-statistic:                     68.95
Date:                Mon, 02 Dec 2024   Prob (F-statistic):           3.15e-12
Time:                        20:59:09   Log-Likelihood:                -377.87
No. Observations:                  77   AIC:                             759.7
Df Residuals:                      75   BIC:                             764.4
Df Model:                           1
Covariance Type:            nonrobust
```

```
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          134.5013      8.131     16.542      0.000     118.304     150.699
unemployment     4.4932      0.541      8.304      0.000       3.415       5.571
==============================================================================
Omnibus:                        1.059   Durbin-Watson:                   1.509
Prob(Omnibus):                  0.589   Jarque-Bera (JB):                1.120
Skew:                          -0.202   Prob(JB):                        0.571
Kurtosis:                       2.568   Cond. No.                         32.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:       diabetes_related   R-squared:                       0.415
Model:                            OLS   Adj. R-squared:                  0.407
Method:                 Least Squares   F-statistic:                     53.24
Date:                Mon, 02 Dec 2024   Prob (F-statistic):           2.57e-10
Time:                        20:59:09   Log-Likelihood:                -324.34
No. Observations:                  77   AIC:                             652.7
Df Residuals:                      75   BIC:                             657.4
Df Model:                           1
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025
                          0.975]
---------------------------------------------------------------------------------------
const                  47.4889      3.845     12.351      0.000      39.829
55.149
below_poverty_level     1.2048      0.165      7.297      0.000       0.876
1.534
==============================================================================
Omnibus:                        2.674   Durbin-Watson:                   1.589
Prob(Omnibus):                  0.263   Jarque-Bera (JB):                2.164
Skew:                          -0.158   Prob(JB):                        0.339
Kurtosis:                       3.758   Cond. No.                         47.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:       diabetes_related   R-squared:                       0.076
Model:                            OLS   Adj. R-squared:                  0.063
```

```
Method:                 Least Squares   F-statistic:                     6.132
Date:                Mon, 02 Dec 2024   Prob (F-statistic):             0.0155
Time:                        20:59:09   Log-Likelihood:                -341.96
No. Observations:                  77   AIC:                             687.9
Df Residuals:                      75   BIC:                             692.6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================================
                           coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------------------------------
const                   61.6071      4.799     12.838      0.000      52.048
71.167
no_high_school_diploma    0.4784      0.193      2.476      0.016       0.094
0.863
==============================================================================
Omnibus:                        2.773   Durbin-Watson:                   1.176
Prob(Omnibus):                  0.250   Jarque-Bera (JB):                2.293
Skew:                           0.297   Prob(JB):                        0.318
Kurtosis:                       2.399   Cond. No.                         50.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:       diabetes_related   R-squared:                       0.362
Model:                            OLS   Adj. R-squared:                  0.354
Method:                 Least Squares   F-statistic:                     42.60
Date:                Mon, 02 Dec 2024   Prob (F-statistic):           7.06e-09
Time:                        20:59:09   Log-Likelihood:                -327.67
No. Observations:                  77   AIC:                             659.3
Df Residuals:                      75   BIC:                             664.0
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                       coef    std err          t      P>|t|      [0.025
0.975]
-------------------------------------------------------------------------------------
const               93.6633      3.868     24.217      0.000      85.959
101.368
per_capita_income   -0.0009      0.000     -6.527      0.000      -0.001
-0.001
==============================================================================
Omnibus:                        2.453   Durbin-Watson:                   1.446
Prob(Omnibus):                  0.293   Jarque-Bera (JB):                2.050
Skew:                           0.399   Prob(JB):                        0.359
```

```
Kurtosis:                       3.047   Cond. No.                     5.73e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.73e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
Dep. Variable:       diabetes_related   R-squared:                       0.489
Model:                            OLS   Adj. R-squared:                  0.482
Method:                 Least Squares   F-statistic:                     71.71
Date:                Mon, 02 Dec 2024   Prob (F-statistic):           1.53e-12
Time:                        20:59:09   Log-Likelihood:                -319.16
No. Observations:                  77   AIC:                             642.3
Df Residuals:                      75   BIC:                             647.0
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const           43.5027      3.793     11.470      0.000      35.947      51.058
unemployment     2.1373      0.252      8.468      0.000       1.635       2.640
==============================================================================
Omnibus:                        5.780   Durbin-Watson:                   1.690
Prob(Omnibus):                  0.056   Jarque-Bera (JB):                4.983
Skew:                           0.553   Prob(JB):                       0.0828
Kurtosis:                       3.576   Cond. No.                         32.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:           tuberculosis   R-squared:                       0.131
Model:                            OLS   Adj. R-squared:                  0.119
Method:                 Least Squares   F-statistic:                     11.31
Date:                Mon, 02 Dec 2024   Prob (F-statistic):            0.00121
Time:                        20:59:09   Log-Likelihood:                -220.65
No. Observations:                  77   AIC:                             445.3
Df Residuals:                      75   BIC:                             450.0
Df Model:                           1
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025
                          0.975]
```

```
---------------------------------------------------------------------------------------
const                  3.9124      1.000      3.912      0.000       1.920
5.905
below_poverty_level    0.1445      0.043      3.364      0.001       0.059
0.230
==============================================================================
Omnibus:                        6.979   Durbin-Watson:                   1.666
Prob(Omnibus):                  0.031   Jarque-Bera (JB):                6.661
Skew:                           0.539   Prob(JB):                       0.0358
Kurtosis:                       3.957   Cond. No.                         47.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:           tuberculosis   R-squared:                       0.145
Model:                            OLS   Adj. R-squared:                  0.134
Method:                 Least Squares   F-statistic:                     12.74
Date:                Mon, 02 Dec 2024   Prob (F-statistic):           0.000628
Time:                        20:59:09   Log-Likelihood:                -220.02
No. Observations:                  77   AIC:                             444.0
Df Residuals:                      75   BIC:                             448.7
Df Model:                           1
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------------------
const                     3.7882      0.985      3.847      0.000       1.826
5.750
no_high_school_diploma    0.1415      0.040      3.569      0.001       0.063
0.220
==============================================================================
Omnibus:                        8.775   Durbin-Watson:                   1.569
Prob(Omnibus):                  0.012   Jarque-Bera (JB):                8.352
Skew:                           0.753   Prob(JB):                       0.0154
Kurtosis:                       3.577   Cond. No.                         50.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:           tuberculosis   R-squared:                       0.076
Model:                            OLS   Adj. R-squared:                  0.064
```

```
Method:                 Least Squares   F-statistic:                     6.205
Date:                Mon, 02 Dec 2024   Prob (F-statistic):             0.0149
Time:                        20:59:09   Log-Likelihood:                -223.00
No. Observations:                  77   AIC:                             450.0
Df Residuals:                      75   BIC:                             454.7
Df Model:                           1
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025
0.975]
-------------------------------------------------------------------------------------
const                 8.9737      0.993      9.034      0.000       6.995
10.952
per_capita_income -8.482e-05    3.4e-05     -2.491      0.015      -0.000
-1.7e-05
==============================================================================
Omnibus:                        9.945   Durbin-Watson:                   1.597
Prob(Omnibus):                  0.007   Jarque-Bera (JB):                9.760
Skew:                           0.793   Prob(JB):                      0.00760
Kurtosis:                       3.725   Cond. No.                     5.73e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.73e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
Dep. Variable:           tuberculosis   R-squared:                       0.007
Model:                            OLS   Adj. R-squared:                 -0.007
Method:                 Least Squares   F-statistic:                    0.5023
Date:                Mon, 02 Dec 2024   Prob (F-statistic):              0.481
Time:                        20:59:09   Log-Likelihood:                -225.80
No. Observations:                  77   AIC:                             455.6
Df Residuals:                      75   BIC:                             460.3
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            6.1362      1.128      5.438      0.000       3.888       8.384
unemployment     0.0532      0.075      0.709      0.481      -0.096       0.203
==============================================================================
Omnibus:                       10.391   Durbin-Watson:                   1.635
Prob(Omnibus):                  0.006   Jarque-Bera (JB):               10.336
Skew:                           0.806   Prob(JB):                      0.00570
Kurtosis:                       3.789   Cond. No.                         32.4
```

```
==============================================================================

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

Crate plots with the analysis outcomes.

```
import altair as alt

# 1-1. below_poverty_level ~ cancer_all_sites

x_var_1 = "below_poverty_level"
y_var_1 = "cancer_all_sites"

# Altair point plot
scatter_plot_1 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_1}:Q", title="Below Poverty Level (%)"),
    y=alt.Y(f"{y_var_1}:Q", title="Cancer (All Sites)"),
    tooltip=[x_var_1, y_var_1]
).properties(
    title=f"Scatter Plot: {x_var_1} vs {y_var_1}",
    width=500,
    height=400
)

# Add line
trend_line_1 = scatter_plot_1.transform_regression(
    x_var_1, y_var_1, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_1 = scatter_plot_1 + trend_line_1

final_chart_1


# 1-2. no_high_school_diploma ~ cancer_all_sites

x_var_2 = "no_high_school_diploma"
y_var_2 = "cancer_all_sites"

# Altair point plot
scatter_plot_2 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_2}:Q", title="No High School Diploma"),
```

```
    y=alt.Y(f"{y_var_2}:Q", title="Cancer (All Sites)"),
    tooltip=[x_var_2, y_var_2]
).properties(
    title=f"Scatter Plot: {x_var_2} vs {y_var_2}",
    width=500,
    height=400
)

# Add line
trend_line_2 = scatter_plot_2.transform_regression(
    x_var_2, y_var_2, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_2 = scatter_plot_2 + trend_line_2

final_chart_2


# 1-3. per_capita_income ~ cancer_all_sites

x_var_3 = "per_capita_income"
y_var_3 = "cancer_all_sites"

# Altair point plot
scatter_plot_3 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_3}:Q", title="Per Capita Income"),
    y=alt.Y(f"{y_var_3}:Q", title="Cancer (All Sites)"),
    tooltip=[x_var_3, y_var_3]
).properties(
    title=f"Scatter Plot: {x_var_3} vs {y_var_3}",
    width=500,
    height=400
)

# Add line
trend_line_3 = scatter_plot_3.transform_regression(
    x_var_3, y_var_3, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_3 = scatter_plot_3 + trend_line_3

final_chart_3


# 1-4. unemployment ~ cancer_all_sites
```

```
x_var_4 = "unemployment"
y_var_4 = "cancer_all_sites"

# Altair point plot
scatter_plot_4 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_4}:Q", title="Unemployment"),
    y=alt.Y(f"{y_var_4}:Q", title="Cancer (All Sites)"),
    tooltip=[x_var_4, y_var_4]
).properties(
    title=f"Scatter Plot: {x_var_4} vs {y_var_4}",
    width=500,
    height=400
)

# Add line
trend_line_4 = scatter_plot_4.transform_regression(
    x_var_4, y_var_4, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_4 = scatter_plot_4 + trend_line_4

final_chart_4


# 2-1. below_poverty_level ~ diabetes_related

x_var_5 = "below_poverty_level"
y_var_5 = "diabetes_related"

# Altair point plot
scatter_plot_5 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_5}:Q", title="Below Poverty Level (%)"),
    y=alt.Y(f"{y_var_5}:Q", title="Diabetes Related"),
    tooltip=[x_var_5, y_var_5]
).properties(
    title=f"Scatter Plot: {x_var_5} vs {y_var_5}",
    width=500,
    height=400
)

# Add line
trend_line_5 = scatter_plot_5.transform_regression(
    x_var_5, y_var_5, method="linear"
).mark_line(color="red")
```

```python
# point plot + line
final_chart_5 = scatter_plot_5 + trend_line_5

final_chart_5


# 2-2. no_high_school_diploma ~ diabetes_related

x_var_6 = "no_high_school_diploma"
y_var_6 = "diabetes_related"

# Altair point plot
scatter_plot_6 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_6}:Q", title="No High School Diploma"),
    y=alt.Y(f"{y_var_6}:Q", title="Diabetes Related"),
    tooltip=[x_var_6, y_var_6]
).properties(
    title=f"Scatter Plot: {x_var_6} vs {y_var_6}",
    width=500,
    height=400
)

# Add line
trend_line_6 = scatter_plot_6.transform_regression(
    x_var_6, y_var_6, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_6 = scatter_plot_6 + trend_line_6

final_chart_6


# 2-3. per_capita_income ~ diabetes_related

x_var_7 = "per_capita_income"
y_var_7 = "diabetes_related"

# Altair point plot
scatter_plot_7 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_7}:Q", title="Per Capita Income"),
    y=alt.Y(f"{y_var_7}:Q", title="Diabetes Related"),
    tooltip=[x_var_7, y_var_7]
).properties(
    title=f"Scatter Plot: {x_var_7} vs {y_var_7}",
    width=500,
    height=400
```

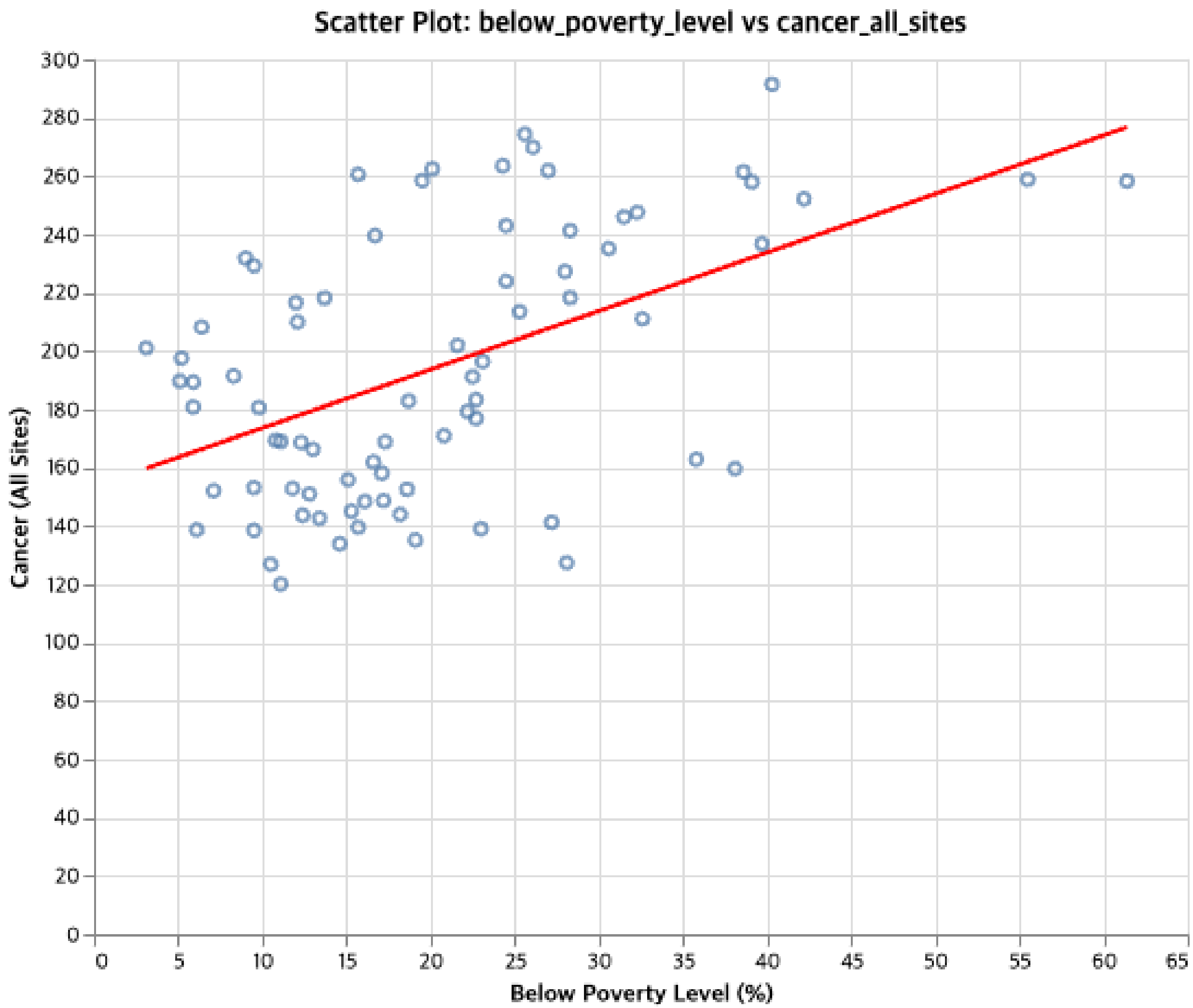
```
)

# Add line
trend_line_7 = scatter_plot_7.transform_regression(
    x_var_7, y_var_7, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_7 = scatter_plot_7 + trend_line_7

final_chart_7


# 2-4. unemployment ~ diabetes_related

x_var_8 = "unemployment"
y_var_8 = "diabetes_related"

# Altair point plot
scatter_plot_8 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_8}:Q", title="Unemployment"),
    y=alt.Y(f"{y_var_8}:Q", title="Diabetes Related"),
    tooltip=[x_var_8, y_var_8]
).properties(
    title=f"Scatter Plot: {x_var_8} vs {y_var_8}",
    width=500,
    height=400
)

# Add line
trend_line_8 = scatter_plot_8.transform_regression(
    x_var_8, y_var_8, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_8 = scatter_plot_8 + trend_line_8

final_chart_8


# 3-1. below_poverty_level ~ tuberculosis

x_var_9 = "below_poverty_level"
y_var_9 = "tuberculosis"

# Altair point plot
scatter_plot_9 = alt.Chart(data).mark_point().encode(
```

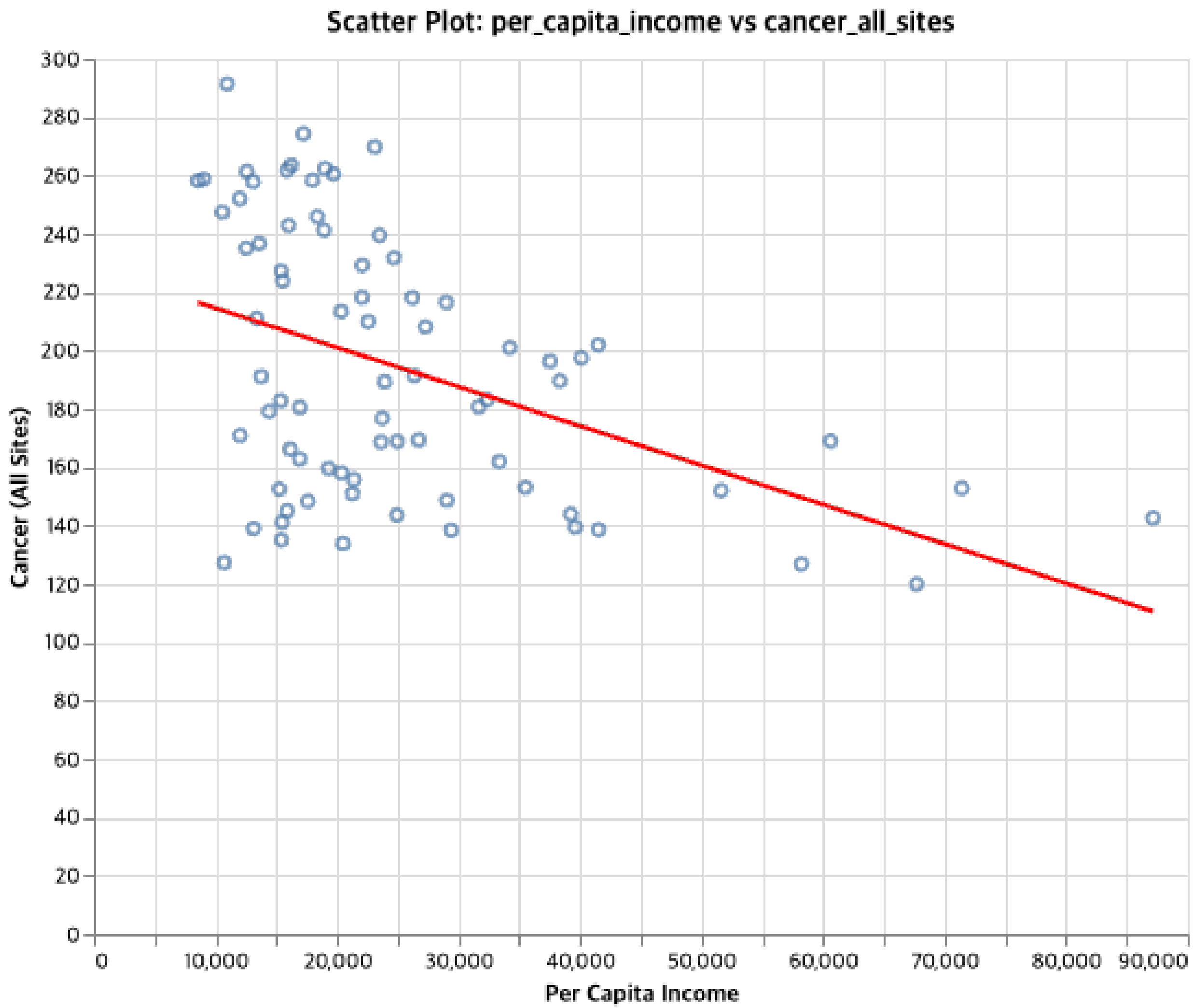
```
    x=alt.X(f"{x_var_9}:Q", title="Below Poverty Level (%)"),
    y=alt.Y(f"{y_var_9}:Q", title="Tuberculosis"),
    tooltip=[x_var_9, y_var_9]
).properties(
    title=f"Scatter Plot: {x_var_9} vs {y_var_9}",
    width=500,
    height=400
)

# Add line
trend_line_9 = scatter_plot_9.transform_regression(
    x_var_9, y_var_9, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_9 = scatter_plot_9 + trend_line_9

final_chart_9


# 3-2. no_high_school_diploma ~ tuberculosis

x_var_10 = "no_high_school_diploma"
y_var_10 = "tuberculosis"

# Altair point plot
scatter_plot_10 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_10}:Q", title="No High School Diploma"),
    y=alt.Y(f"{y_var_10}:Q", title="Tuberculosis"),
    tooltip=[x_var_10, y_var_10]
).properties(
    title=f"Scatter Plot: {x_var_10} vs {y_var_10}",
    width=500,
    height=400
)

# Add line
trend_line_10 = scatter_plot_10.transform_regression(
    x_var_10, y_var_10, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_10 = scatter_plot_10 + trend_line_10

final_chart_10
```

```
# 3-3. per_capita_income ~ tuberculosis

x_var_11 = "per_capita_income"
y_var_11 = "tuberculosis"

# Altair point plot
scatter_plot_11 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_11}:Q", title="Per Capita Income"),
    y=alt.Y(f"{y_var_11}:Q", title="Tuberculosis"),
    tooltip=[x_var_11, y_var_11]
).properties(
    title=f"Scatter Plot: {x_var_11} vs {y_var_11}",
    width=500,
    height=400
)

# Add line
trend_line_11 = scatter_plot_11.transform_regression(
    x_var_11, y_var_11, method="linear"
).mark_line(color="red")

# point plot + line
final_chart_11 = scatter_plot_11 + trend_line_11

final_chart_11


# 3-4. unemployment ~ tuberculosis

x_var_12 = "unemployment"
y_var_12 = "tuberculosis"

# Altair point plot
scatter_plot_12 = alt.Chart(data).mark_point().encode(
    x=alt.X(f"{x_var_12}:Q", title="Unemployment"),
    y=alt.Y(f"{y_var_12}:Q", title="Tuberculosis"),
    tooltip=[x_var_12, y_var_12]
).properties(
    title=f"Scatter Plot: {x_var_12} vs {y_var_12}",
    width=500,
    height=400
)

# Add line
trend_line_12 = scatter_plot_12.transform_regression(
    x_var_12, y_var_12, method="linear"
).mark_line(color="red")
```

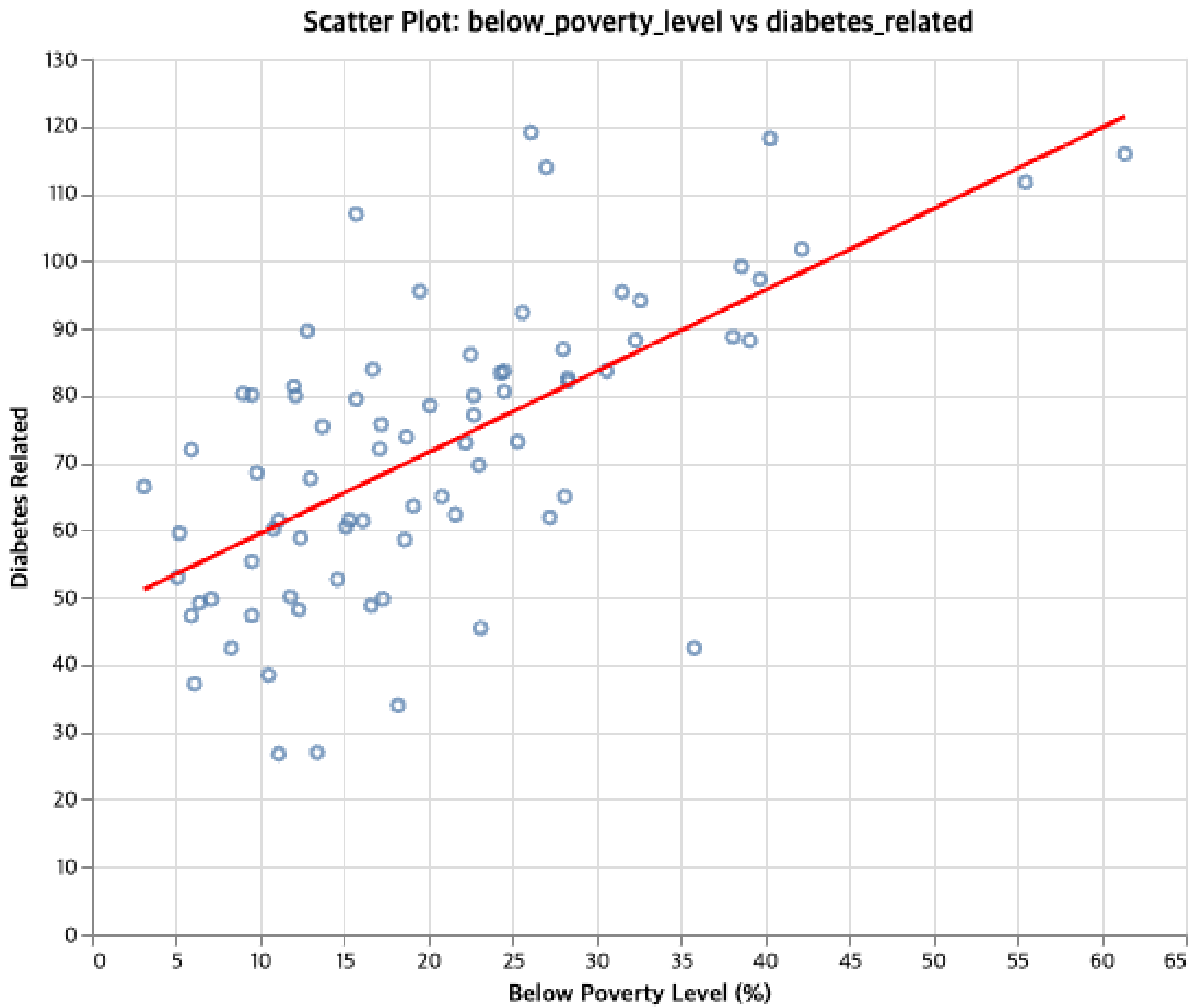
```
# point plot + line
final_chart_12 = scatter_plot_12 + trend_line_12

final_chart_12
```

```
alt.LayerChart(...)
```

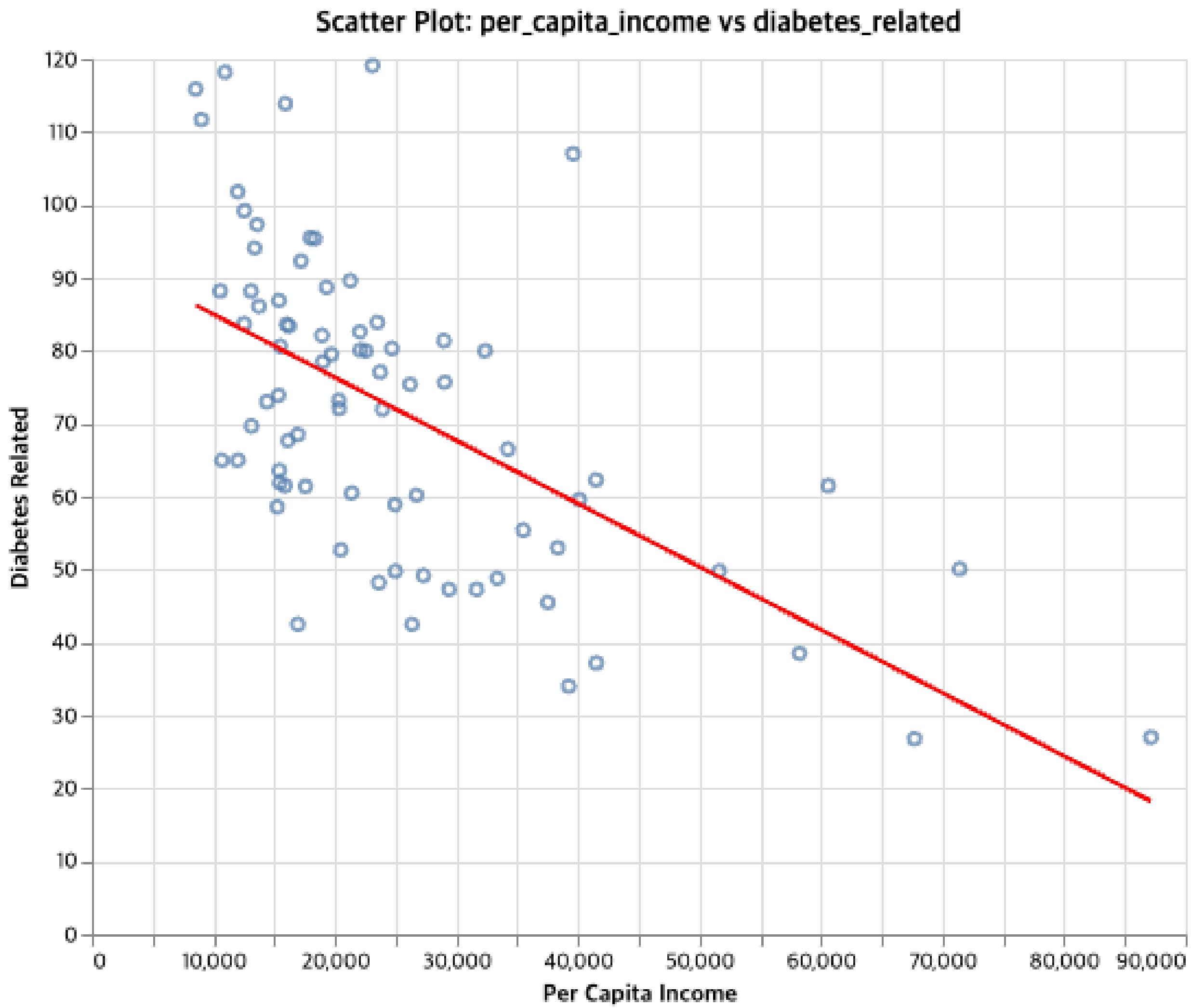


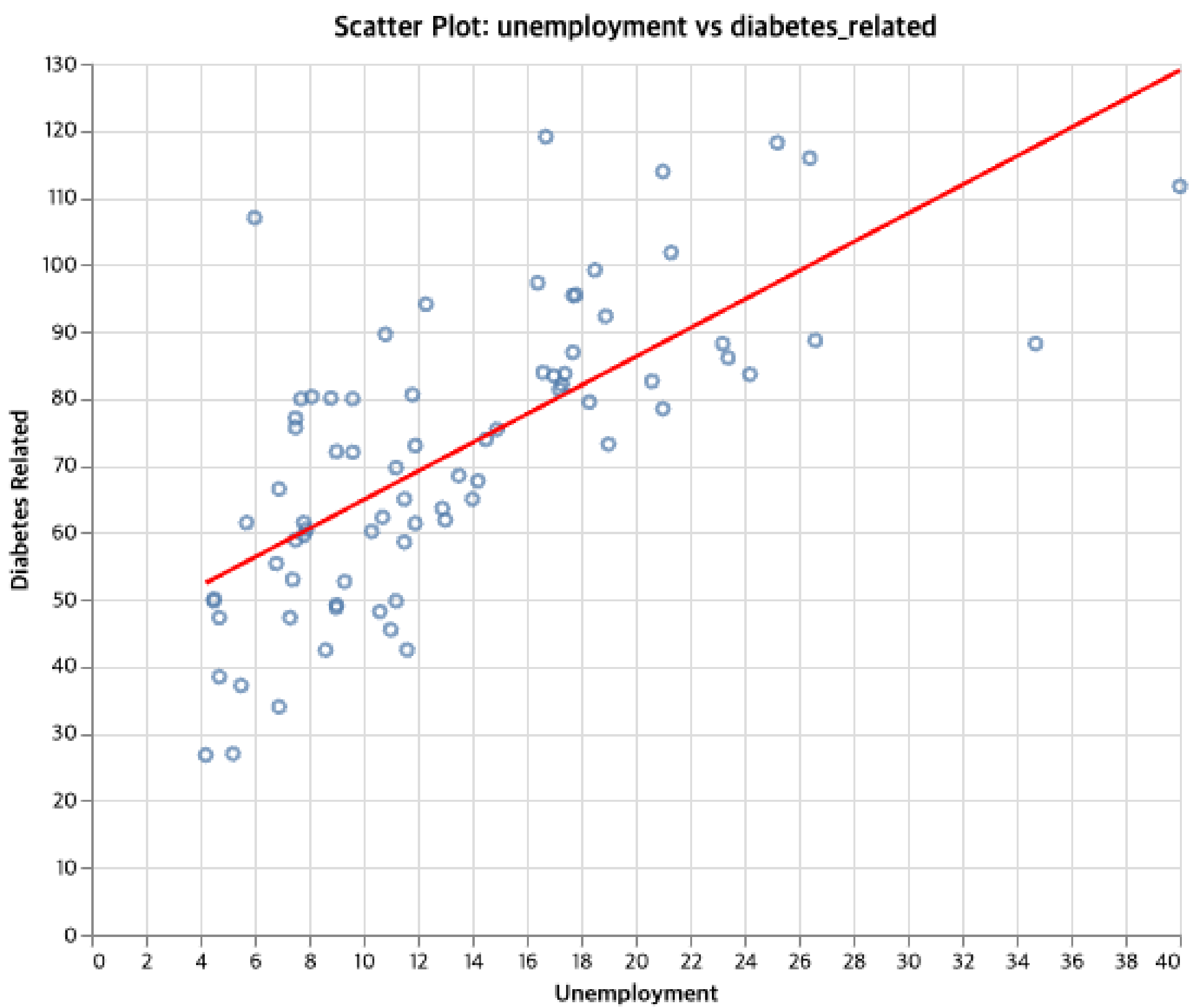

Scatter Plot: per_capita_income vs cancer_all_sites
Cancer (All Sites)
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
Per Capita Income

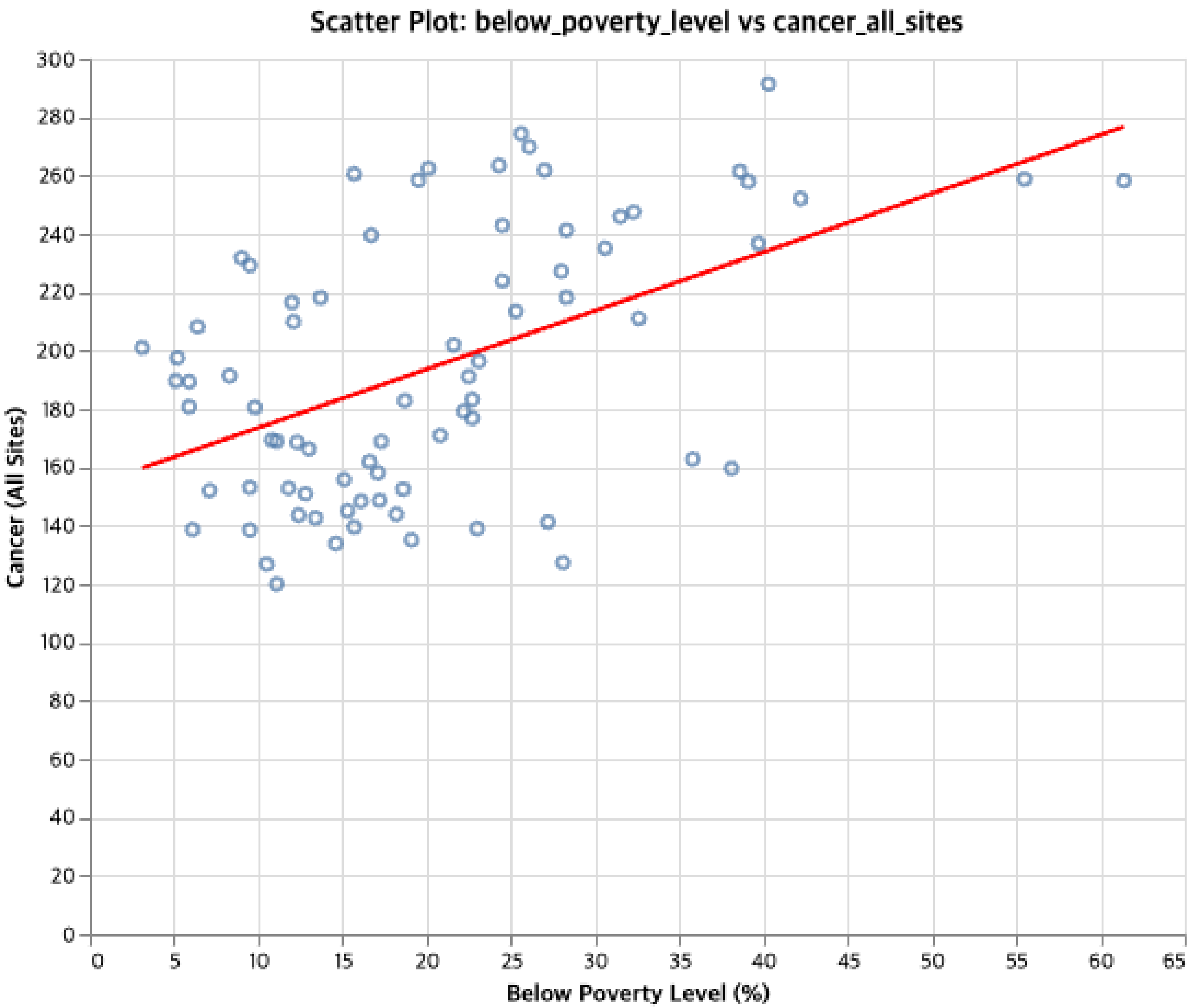
Scatter Plot: unemployment vs cancer_all_sites
Cancer (All Sites)
Unemployment
350
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40

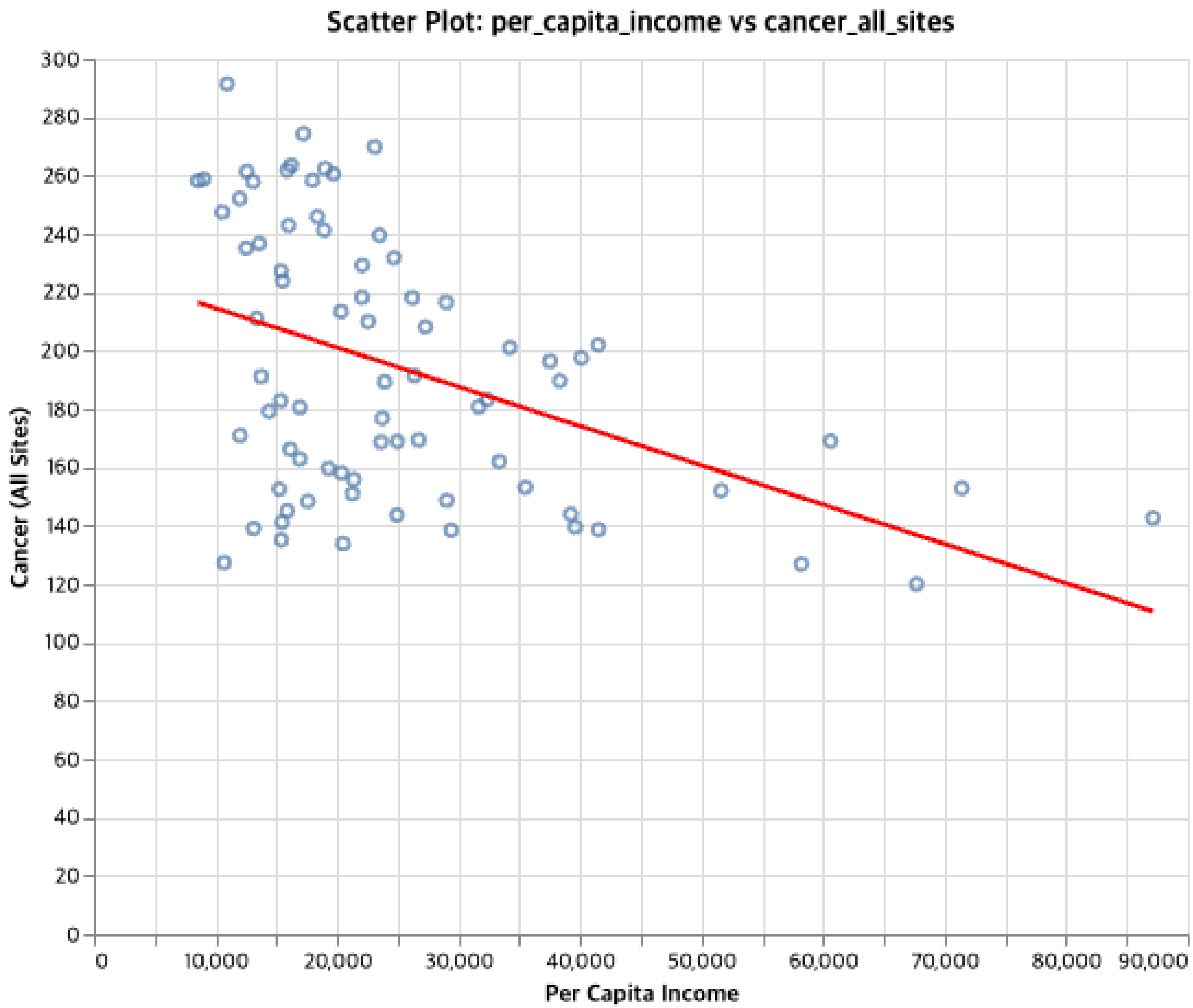


Scatter Plot: below_poverty_level vs diabetes_related
Diabetes Related
Below Poverty Level (%)
0
10
20
30
40
50
60
70
80
90
100
110
120
130
0
5
10
15
20
25
30
35
40
45
50
55
60
65


Scatter Plot: per_capita_income vs diabetes_related
Diabetes Related
Per Capita Income
0
10
20
30
40
50
60
70
80
90
100
110
120
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000

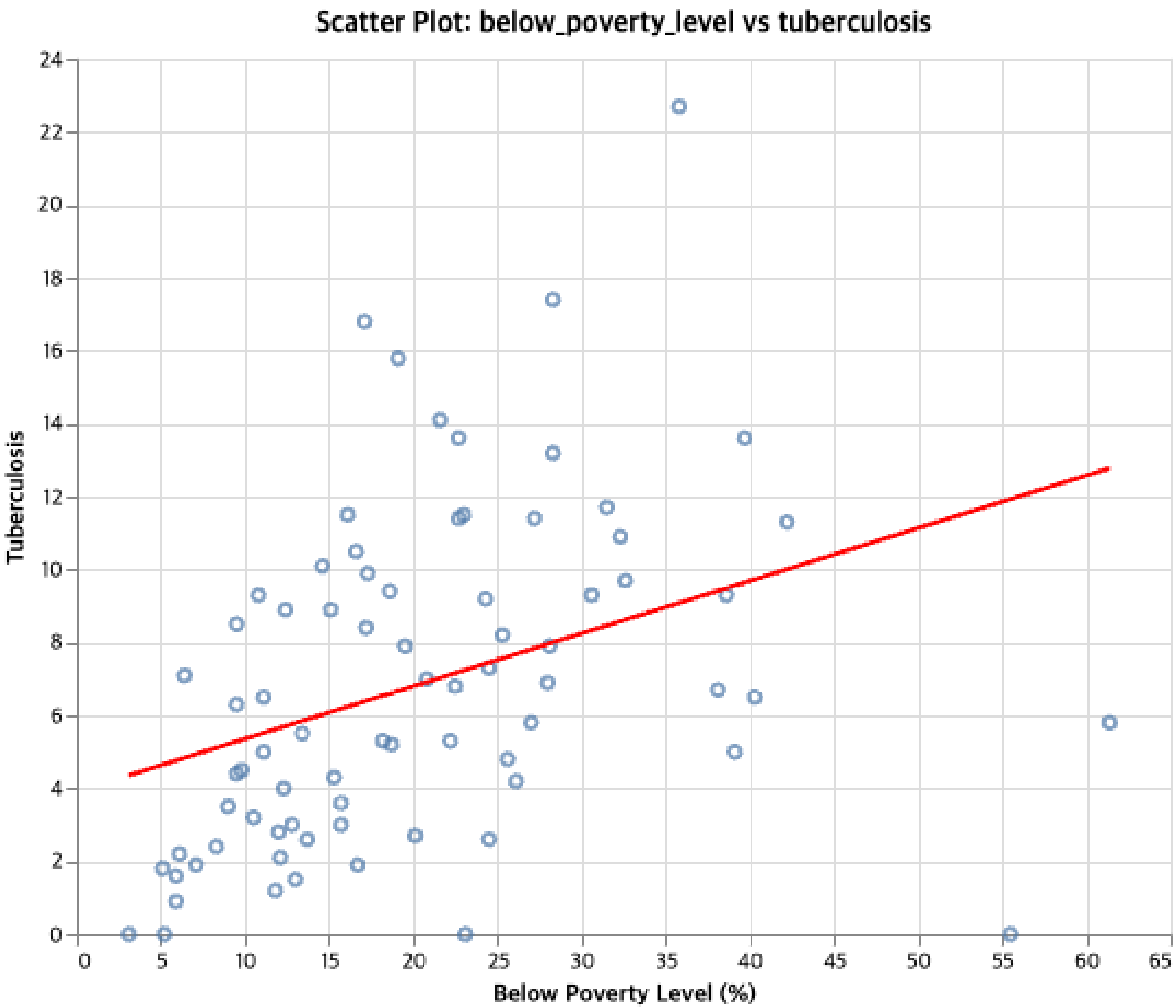

Scatter Plot: unemployment vs diabetes_related
Diabetes Related
Unemployment
130
120
110
100
90
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40

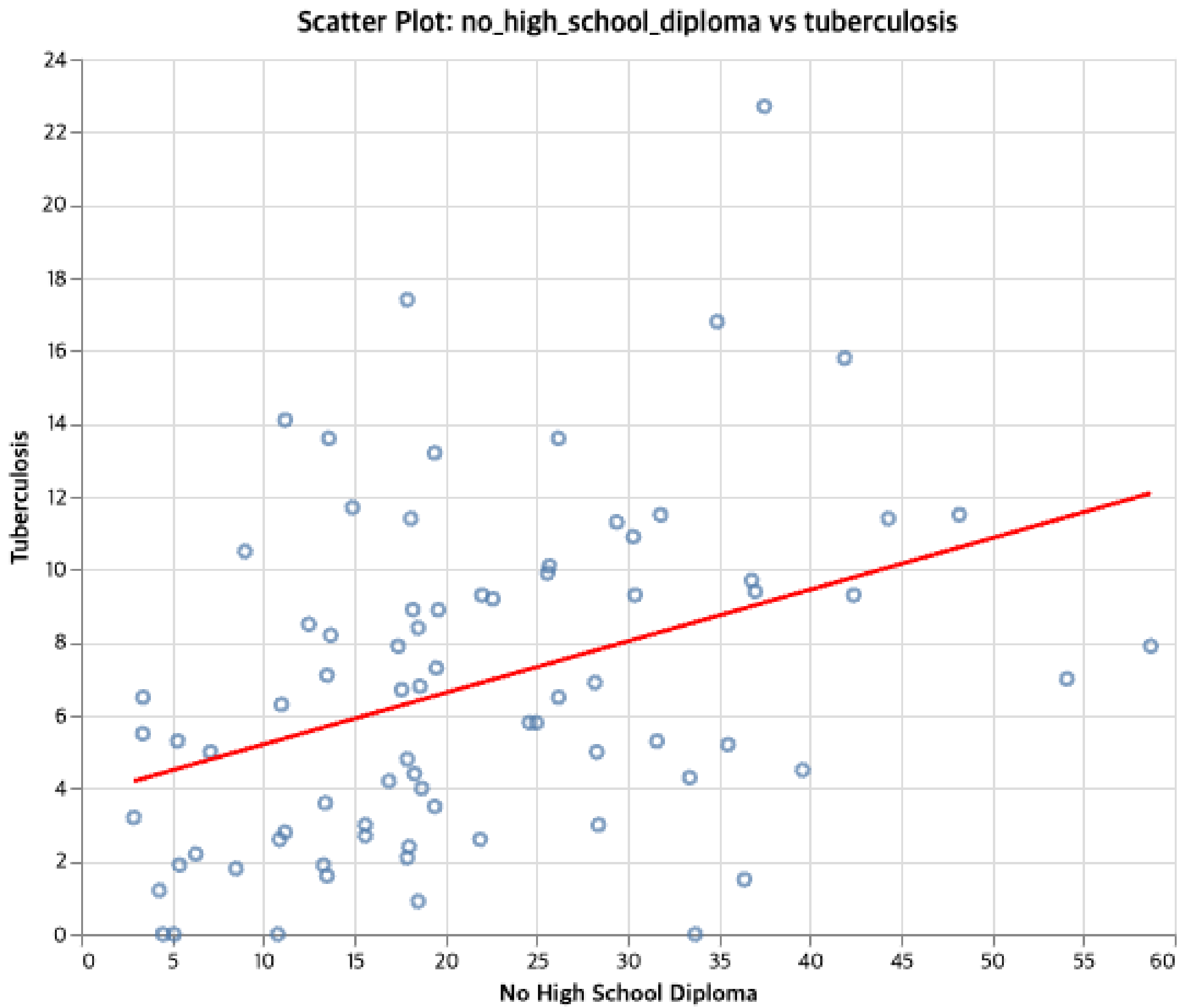

Scatter Plot: below_poverty_level vs cancer_all_sites
Cancer (All Sites)
Below Poverty Level (%)
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
0
5
10
15
20
25
30
35
40
45
50
55
60
65



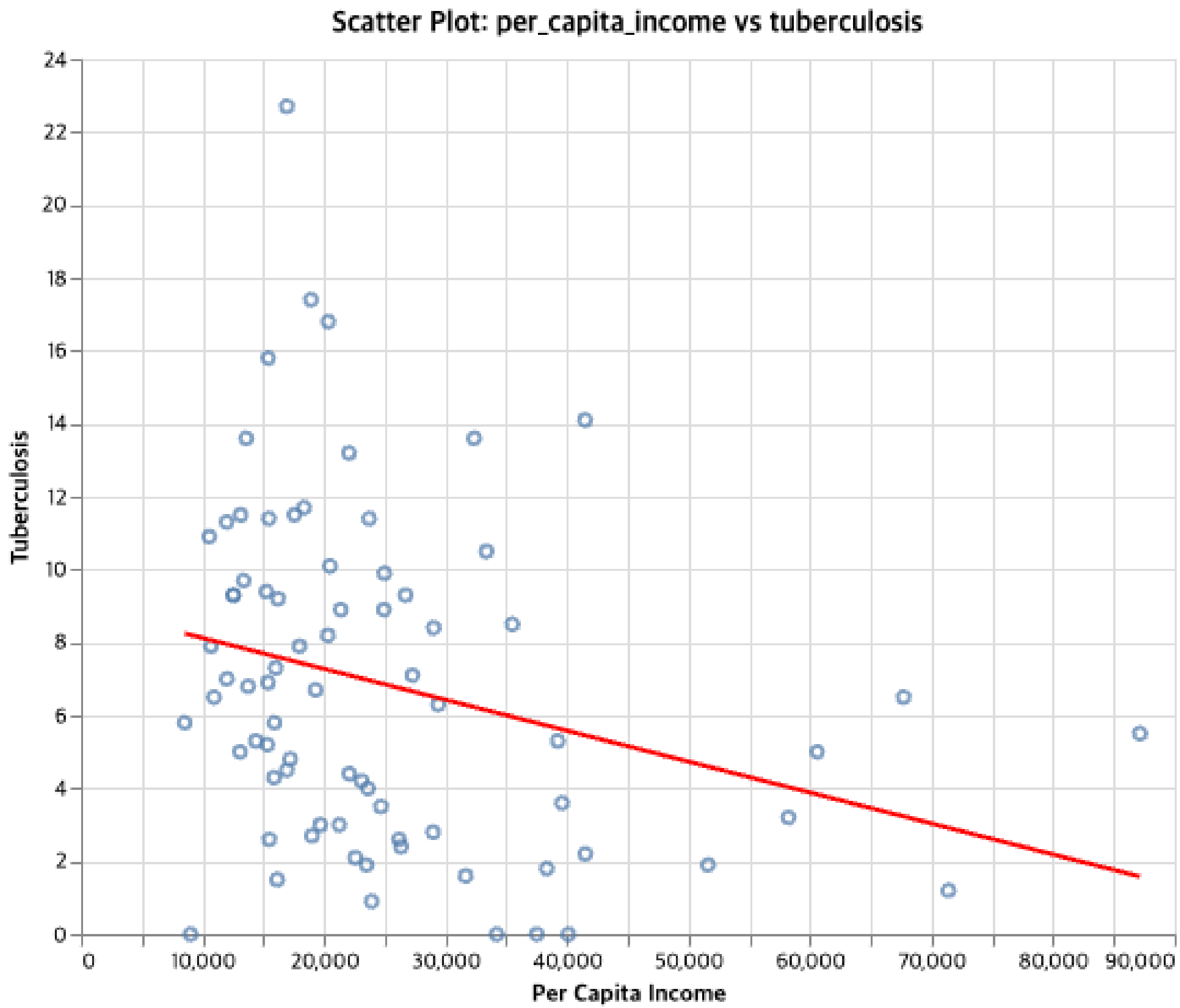
Scatter Plot: per_capita_income vs cancer_all_sites
Cancer (All Sites)
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
0
10,000
20,000
30,000
40,000
50,000
60,000
70,000
80,000
90,000
Per Capita Income

Scatter Plot: unemployment vs cancer_all_sites
Cancer (All Sites)
Unemployment
0
50
100
150
200
250
300
350
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40


Scatter Plot: below_poverty_level vs tuberculosis
Tuberculosis
Below Poverty Level (%)
0
2
4
6
8
10
12
14
16
18
20
22
24
0
5
10
15
20
25
30
35
40
45
50
55
60
65



Scatter Plot: no_high_school_diploma vs tuberculosis
Tuberculosis
No High School Diploma
0
2
4
6
8
10
12
14
16
18
20
22
24
0
5
10
15
20
25
30
35
40
45
50
55
60



3. Text Analysis

url: https://www.cbsnews.com/chicago/news/cancer-care-disparities-1/

Subject: Chicago's South, West Sides have many more cancer patients, less access to care.

Scrape the article content

```
import requests
from bs4 import BeautifulSoup
```

```python
# URL
url = 'https://www.cbsnews.com/chicago/news/cancer-care-disparities-1/'

# Load the web page
response = requests.get(url)
response.raise_for_status()  # Check if the request is successful

# HTML parsing
soup = BeautifulSoup(response.text, 'html.parser')

#       ( : <div>       'content'   )
article_section = soup.find('section', class_='content__body')
if article_section:
    paragraphs = article_section.find_all('p')
    article_text = '\n'.join([para.get_text() for para in paragraphs])
    print(article_text)
else:
    print("Article is not found.")
```

```
CHICAGO (CBS) - Chicago has long been a hub for breakthroughs in medicine, but
the number of people dying from cancer in the city shows not everyone benefits
equally.
For those in Chicago's low-income, predominately Black or Latino neighborhoods,
they don't have the same access to or quality of care. CBS 2's Audrina Sinclair
examined the problem and who's helping.
From Genella Jones-Riggins' backyard in Roseland, she grows it and cans it.
"[There are] eight or nine different types of tomatoes," she said.
She has stacks and stacks of jars of "shelf-ready food" for at least six months
to a year, she said. It's piles of produce.
"It's wholesome food that I grew myself," Jones-Riggins  said. "It's my
responsibility to make sure that I stay healthy."
Speaking of health…
Sinclair: "Where are you at now in your journey?"
Jones-Riggins: "My prognosis is well. My last scans were clean."
The good news came a year after finding a lump in her breast and having no health
insurance.
"You hear all the time about breast cancer, free screenings, but when I needed
it, I couldn't find it," Jones-Riggins  said. "I called around for two weeks, and
I could not find anything."
A friend told her about the nonprofit Equal Hope.
"They made sure that I got everything that I needed," she said.
That started with a mammogram and biopsy at Rush University Medical Center, where
doctors diagnosed her with triple negative breast cancer.
Sinclair: "Someone went to every appointment with you. How many appointments are
we talking about?"
```

```
Jones-Riggins: "I had 17 rounds of chemo. MRIs, and you have CT scans, and you
have bone scans, and you have bone density scans. Then after chemo, you have
radiation and radiation is every single day for four to five weeks."
Her nurse navigator, Rita, was there for her, and the costs were all covered.
"This is a state-funded program that allows women to access care at no cost,"
said Paris Thomas, of Equal Hope. "Which is why she didn't have to see those
bills."
Thomas works to fight Chicago's cancer care disparities with Equal Hope, which
helps 1,800 women like Jones-Riggins with breast or cervical cancer.
"We serve the communities on the West and South Side of Chicago," Thomas said.
"Primarily those that are Black and brown, and usually those who are considered
under-resourced, disinvested in."
A map from the Chicago Department of Public Health and PHAME Center at the
University of Illinois at Chicago shows the neighborhoods where the most people
are dying from cancer in Chicago. The darker the blue, the more cancer deaths.
All but one of the communities are on the South and West Sides.
"We know that we have to intervene in this community because there's a problem
here," Thomas said.
Equal Hope is intervening for patients treated at safety net hospitals in their
neighborhoods. Such hospitals are usually under-resourced facilities with
outdated equipment, lower staffing levels, and limited hours.
"Let's say capacity," said Thomas. "Maybe they don't have a full-time mammogram
tech, and they're only able to see women once a week. So we know that we now have
to pivot and try to move our populations to other facilities."
A study by the Health Care Council of Chicago looked at those barriers to care in
the city's under-resourced neighborhoods and found specialists on the South Side
are treating three times as many patients as on the North Side. That's about
1,000 patients for every doctor on the South Side of the city, compared to about
350 patients for every doctor in many North Side communities.
Thomas is hyper-focused on the disparities to help people like Jones-Riggins  get
the cancer care they need.
"I don't know where I'd be without the help that they provided me," said a
tearful Jones-Riggins.
To learn more about Equal Hope and its services, visit EqualHope.org.
In the second part of her story, Sinclair will dig deeper into the disparities
and ways to tackle them, including a look at a new cancer center coming to Hyde
Park.
Audrina Sinclair is an anchor on the CBS2 Morning News.
```

Text Analysis - semtimental

```python
from textblob import TextBlob

# Create textblob
blob = TextBlob(article_text)
```

```
# Sentiment Analysis
sentiment = blob.sentiment
polarity = sentiment.polarity  # -1(negative) +1(oisitive)
subjectivity = sentiment.subjectivity  # 0(objective) 1(subjective)

print(f"Polarity: {polarity}")  # 0.10 (slightly positive)
print(f"Subjectivity: {subjectivity}")  # 0.38 (relatively objective)
```

```
Polarity: 0.10115199615199617
Subjectivity: 0.38106220939554264
```

Text Analysis - semtimental by sentence

```
import spacy
nlp = spacy.load("en_core_web_sm")

doc = nlp(article_text)
type(doc)

sents = list(doc.sents)
sents_list = list(doc.sents)
sents_list

for token in sents_list[1]:
    print(token.text)
```

```
For
those
in
Chicago
's
low
-
income
,
predominately
Black
or
Latino
neighborhoods
,
they
do
```

```
n't
have
the
same
access
to
or
quality
of
care
.
```

```
from textblob import TextBlob
import pandas as pd
import altair as alt

# Calculate setence polarity
sentence_polarities = []
for i, sentence in enumerate(sents_list):
    blob = TextBlob(sentence.text)
    polarity = blob.sentiment.polarity
    sentence_polarities.append({"n": i + 1, "sentence": sentence.text,
↪   "polarity": polarity})

# Create a DataFrame
df_polarity = pd.DataFrame(sentence_polarities)

# Print the outcome
print(df_polarity)
```

```
     n                                           sentence  polarity
0    1  CHICAGO (CBS) - Chicago has long been a hub fo... -0.025000
1    2  For those in Chicago's low-income, predominate... -0.083333
2    3  CBS 2's Audrina Sinclair examined the problem ...  0.000000
3    4  From Genella Jones-Riggins' backyard in Rosela...  0.000000
4    5  "[There are] eight or nine different types of ...  0.000000
5    6  She has stacks and stacks of jars of "shelf-re... -0.300000
6    7                          It's piles of produce.\n  0.000000
7    8  "It's wholesome food that I grew myself," Jone...  0.000000
8    9  "It's my responsibility to make sure that I st...  0.500000
9   10  Speaking of health…\nSinclair: "Where are you ...  0.000000
10  11                          My last scans were clean.  0.183333
11  12  "\nThe good news came a year after finding a l...  0.700000
12  13  "You hear all the time about breast cancer, fr...  0.400000
13  14  "I called around for two weeks, and I could no...  0.000000
14  15  A friend told her about the nonprofit Equal Ho...  0.000000
15  16  "They made sure that I got everything that I n...  0.500000
```

```
16  17  That started with a mammogram and biopsy at Ru... -0.133333
17  18  Sinclair: "Someone went to every appointment w...  0.000000
18  19  How many appointments are we talking about?"\n...  0.500000
19  20  MRIs, and you have CT scans, and you have bone...  0.000000
20  21  Then after chemo, you have radiation and radia... -0.071429
21  22  Her nurse navigator, Rita, was there for her, ...  0.000000
22  23  "This is a state-funded program that allows wo...  0.000000
23  24  "Which is why she didn't have to see those bills.  0.000000
24  25  "\nThomas works to fight Chicago's cancer care...  0.000000
25  26  "We serve the communities on the West and Sout...  0.000000
26  27  "Primarily those that are Black and brown, and... -0.005556
27  28  "\nA map from the Chicago Department of Public...  0.133333
28  29     The darker the blue, the more cancer deaths.\n  0.250000
29  30  All but one of the communities are on the Sout...  0.000000
30  31  "We know that we have to intervene in this com...  0.000000
31  32  Equal Hope is intervening for patients treated...  0.000000
32  33  Such hospitals are usually under-resourced fac... -0.180357
33  34                 "Let's say capacity," said Thomas.  0.000000
34  35  "Maybe they don't have a full-time mammogram t...  0.250000
35  36  So we know that we now have to pivot and try t... -0.125000
36  37  "\nA study by the Health Care Council of Chica...  0.500000
37  38  That's about 1,000 patients for every doctor o...  0.500000
38  39  Thomas is hyper-focused on the disparities to ...  0.000000
39  40  "I don't know where I'd be without the help th...  0.000000
40  41  To learn more about Equal Hope and its service...  0.250000
41  42  In the second part of her story, Sinclair will...  0.012121
42  43  Audrina Sinclair is an anchor on the CBS2 Morn...  0.000000
```

```python
# Create a graph about polarity by sentence
chart_polarity = alt.Chart(df_polarity).mark_line().encode(
    x=alt.X('n:Q', title='Sentence Number'),
    y=alt.Y('polarity:Q', title='Polarity'),
    tooltip=['sentence', 'polarity']
).properties(
    title='Sentence Polarity of Article',
    width=800,
    height=400
).interactive()

chart_polarity
```

```
alt.Chart(...)
```

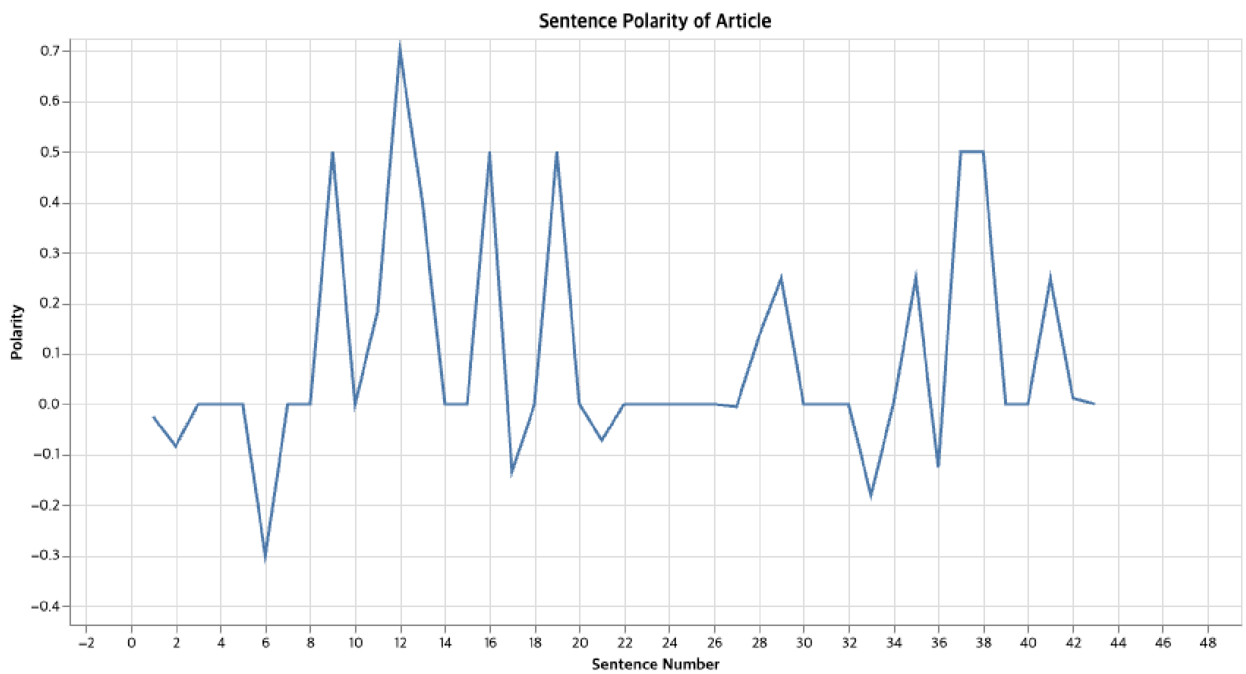
Text Analysis - semtimental by vocabulary

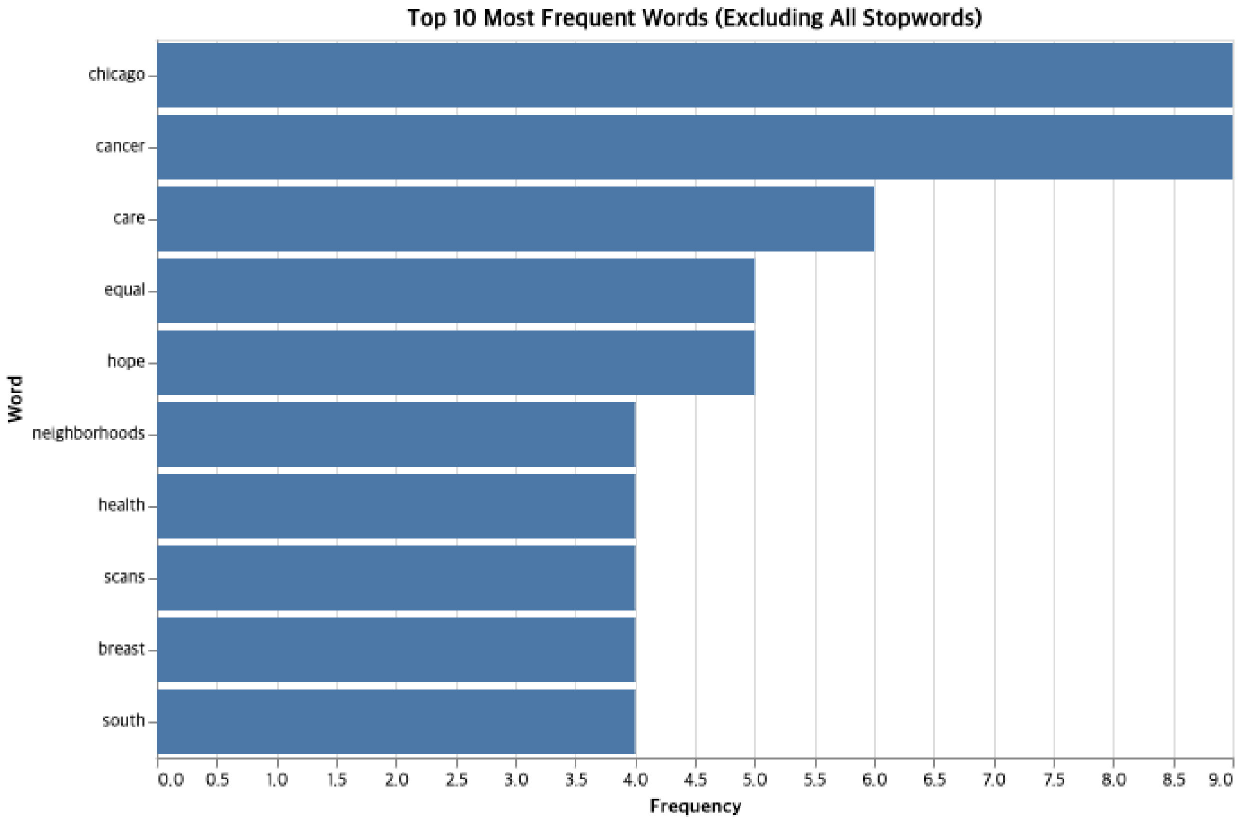
```python
from collections import Counter
import pandas as pd
import altair as alt
import spacy

# Load spaCy model
nlp = spacy.load("en_core_web_sm")

# Define custom stopwords
custom_stopwords = {"said", "jones", "riggins", "thomas", "sinclair"}

# Add custom stopwords to the basic stopwords in spaCy
all_stopwords = nlp.Defaults.stop_words.union(custom_stopwords)

# Remove stopwords and extract words
words = [
    token.text.lower()
    for sentence in sents_list
    for token in sentence
    if token.is_alpha and token.text.lower() not in all_stopwords
]

# Calculate words frequency
word_freq = Counter(words)

# Extract top 10 words
most_common_words = word_freq.most_common(10)
df_word_freq = pd.DataFrame(most_common_words, columns=["word", "frequency"])

# Print the outcome
print(df_word_freq)
```

```
            word  frequency
0        chicago          9
1         cancer          9
2           care          6
3          equal          5
4           hope          5
5  neighborhoods          4
6         health          4
7          scans          4
8         breast          4
9          south          4
```

```
# Create a bar graph
chart_word_frequency = alt.Chart(df_word_freq).mark_bar().encode(
    x=alt.X("frequency:Q", title="Frequency"),
    y=alt.Y("word:N", sort="-x", title="Word"),
    tooltip=["word", "frequency"]
).properties(
    title="Top 10 Most Frequent Words (Excluding All Stopwords)",
    width=600,
    height=400
)

chart_word_frequency
```

alt.Chart(...)




Top 10 Most Frequent Words (Excluding All Stopwords)
Word
chicago
cancer
care
equal
hope
neighborhoods
health
scans
breast
south
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
Frequency