Garment Manufacturing Worker Productivity Factors Analysis

Rachel Chen
Feb 2, 2024

1. Data processing – 10 points.

[ ]:
```
import pandas as pd
import numpy as np
d = pd.read_csv("/content/garments_worker_productivity.csv")
d.head()
```

[ ]:

	date	quarter	department	day	team	targeted_productivity	\
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	
2	1/1/2015	Quarter1	sweing	Thursday	11	0.80	
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	
4	1/1/2015	Quarter1	sweing	Thursday	6	0.80	

	smv	wip	over_time	incentive	idle_time	idle_men	\
0	26.16	1108.0	7080	98	0.0	0	
1	3.94	NaN	960	0	0.0	0	
2	11.41	968.0	3660	50	0.0	0	
3	11.41	968.0	3660	50	0.0	0	
4	25.90	1170.0	1920	50	0.0	0	

	no_of_style_change	no_of_workers	actual_productivity
0	0	59.0	0.940725
1	0	8.0	0.886500
2	0	30.5	0.800570
3	0	30.5	0.800570
4	0	56.0	0.800382

a. Remove the column 'wip' from the dataset.

[ ]:
```
d.drop(columns=['wip'], inplace=True)
d.head()
```

[ ]:

	date	quarter	department	day	team	targeted_productivity	\
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	
2	1/1/2015	Quarter1	sweing	Thursday	11	0.80	
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	

```
4  1/1/2015  Quarter1      sweing  Thursday     6                  0.80

     smv  over_time  incentive  idle_time  idle_men  no_of_style_change  \
0  26.16       7080         98        0.0         0                   0
1   3.94        960          0        0.0         0                   0
2  11.41       3660         50        0.0         0                   0
3  11.41       3660         50        0.0         0                   0
4  25.90       1920         50        0.0         0                   0

   no_of_workers  actual_productivity
0           59.0             0.940725
1            8.0             0.886500
2           30.5             0.800570
3           30.5             0.800570
4           56.0             0.800382
```

b. Create another variable names 'log_productivity' which is defined as log_prodductivity = log(actual_prouctivity * 100). Store any new variable as an additional column in the original data frame.

[ ]:
```
d["log_productivity"]= np.log(d["actual_productivity"] * 100)
d.head()
```

[ ]:
```
        date   quarter  department       day  team  targeted_productivity  \
0  1/1/2015  Quarter1      sweing  Thursday     8                   0.80
1  1/1/2015  Quarter1  finishing   Thursday     1                   0.75
2  1/1/2015  Quarter1      sweing  Thursday    11                   0.80
3  1/1/2015  Quarter1      sweing  Thursday    12                   0.80
4  1/1/2015  Quarter1      sweing  Thursday     6                   0.80

     smv  over_time  incentive  idle_time  idle_men  no_of_style_change  \
0  26.16       7080         98        0.0         0                   0
1   3.94        960          0        0.0         0                   0
2  11.41       3660         50        0.0         0                   0
3  11.41       3660         50        0.0         0                   0
4  25.90       1920         50        0.0         0                   0

   no_of_workers  actual_productivity  log_productivity
0           59.0             0.940725          4.544066
1            8.0             0.886500          4.484696
2           30.5             0.800570          4.382739
3           30.5             0.800570          4.382739
4           56.0             0.800382          4.382504
```

c. Create another variable called 'log_no_of_workers' which is the natural logarithm of the no_of_workers.

[ ]:
```
d["log_no_of_workers"]= np.log(d["no_of_workers"])
d.head()
```

[ ]:

	date	quarter	department	day	team	targeted_productivity	\
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	
2	1/1/2015	Quarter1	sweing	Thursday	11	0.80	
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	
4	1/1/2015	Quarter1	sweing	Thursday	6	0.80	

	smv	over_time	incentive	idle_time	idle_men	no_of_style_change	\
0	26.16	7080	98	0.0	0	0	
1	3.94	960	0	0.0	0	0	
2	11.41	3660	50	0.0	0	0	
3	11.41	3660	50	0.0	0	0	
4	25.90	1920	50	0.0	0	0	

	no_of_workers	actual_productivity	log_productivity	log_no_of_workers
0	59.0	0.940725	4.544066	4.077537
1	8.0	0.886500	4.484696	2.079442
2	30.5	0.800570	4.382739	3.417727
3	30.5	0.800570	4.382739	3.417727
4	56.0	0.800382	4.382504	4.025352

d. Convert the following variables to factor variables (category) team, quarter, department, and day.

[ ]:
```
d["team"]=d["team"].astype("category")
d["quarter"]=d["quarter"].astype("category")
d["department"]=d["department"].astype("category")
d["day"]=d["day"].astype("category")
d.dtypes
```

[ ]:
```
date                      object
quarter                 category
department              category
day                     category
team                    category
targeted_productivity    float64
smv                      float64
over_time                  int64
incentive                  int64
idle_time                float64
idle_men                   int64
no_of_style_change         int64
no_of_workers            float64
actual_productivity      float64
```

```
log_productivity          float64
log_no_of_workers         float64
dtype: object
```

e. Create another variable called 'percentage_achivement' which is defined as follows:

percentage_achievment = (actual_productivity – targeted_productivity) / targeted_productivity X 100.

[ ]:
```
percentage_achievment = (d["actual_productivity"] - d["targeted_productivity"])␣
↪/ d["targeted_productivity"] * 100
print(percentage_achievment)
```

```
0       17.590678
1       18.200000
2        0.071311
3        0.071311
4        0.047743
           …
1192   -16.222222
1193   -10.625000
1194    -3.750000
1195   -32.548148
1196   -43.611111
Length: 1197, dtype: float64
```

f. Also for cleaning the variable department, please run the following command (there are some coding errors in the variable department).

levels(d$department)<-c("finishing", "finishing", "sewing")

[ ]:
```
d['department'] = d['department'].replace({'finishing': 'finishing', 'sewing':␣
↪'finishing'})
```

[ ]:
```
print(d.shape)
```

```
(1197, 16)
```

2. Exploratory Analysis – 40 points.

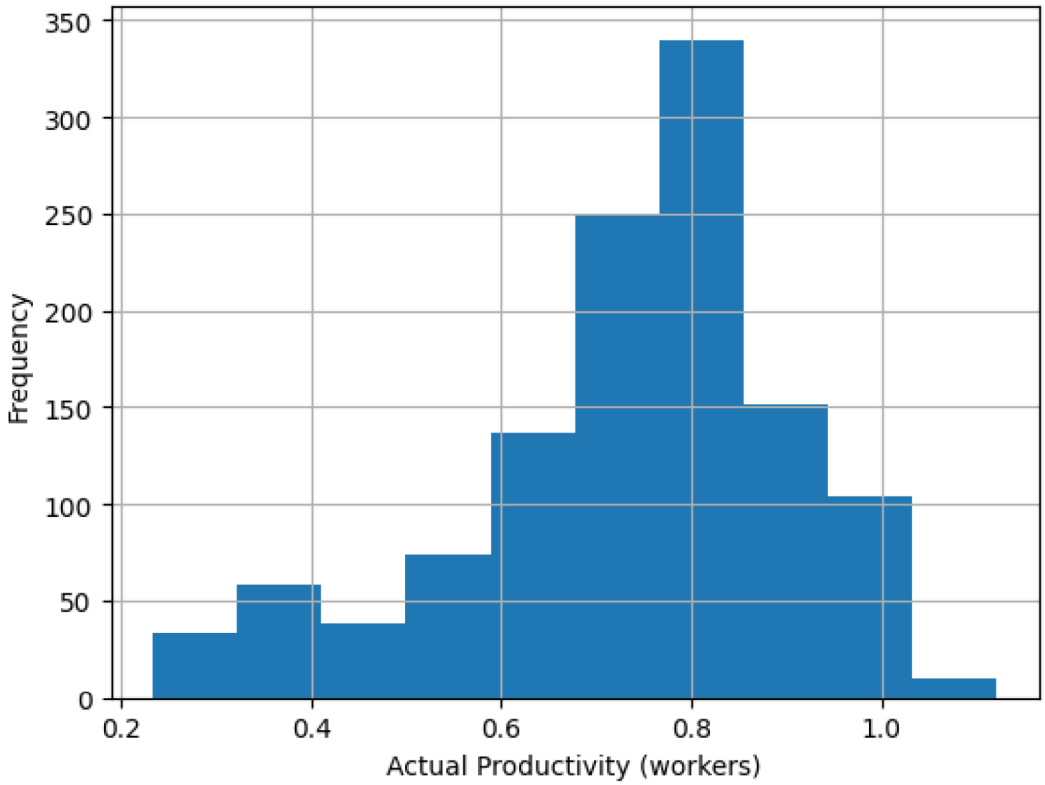
a. Create the histograms of actual_productivity and log_productivity. How does the distribution of log_productivity change with respect to actual_productivity? Do the same for number of workers.

Productivity:

The productivity values become higher when transformed into log_productivity, with the range extending from about 0.23 to 1.12 initially and then widening to approximately 3.15 to 4.72. This broader range indicates greater dispersion in the data. Additionally, there is a higher peak in the log_productivity distribution, suggesting that a significant number of observations fall within a particular range on the logarithmic scale.

Therefore, when data is transformed into logarithmic scale, it tends to become more evenly distributed compared to its original scale.

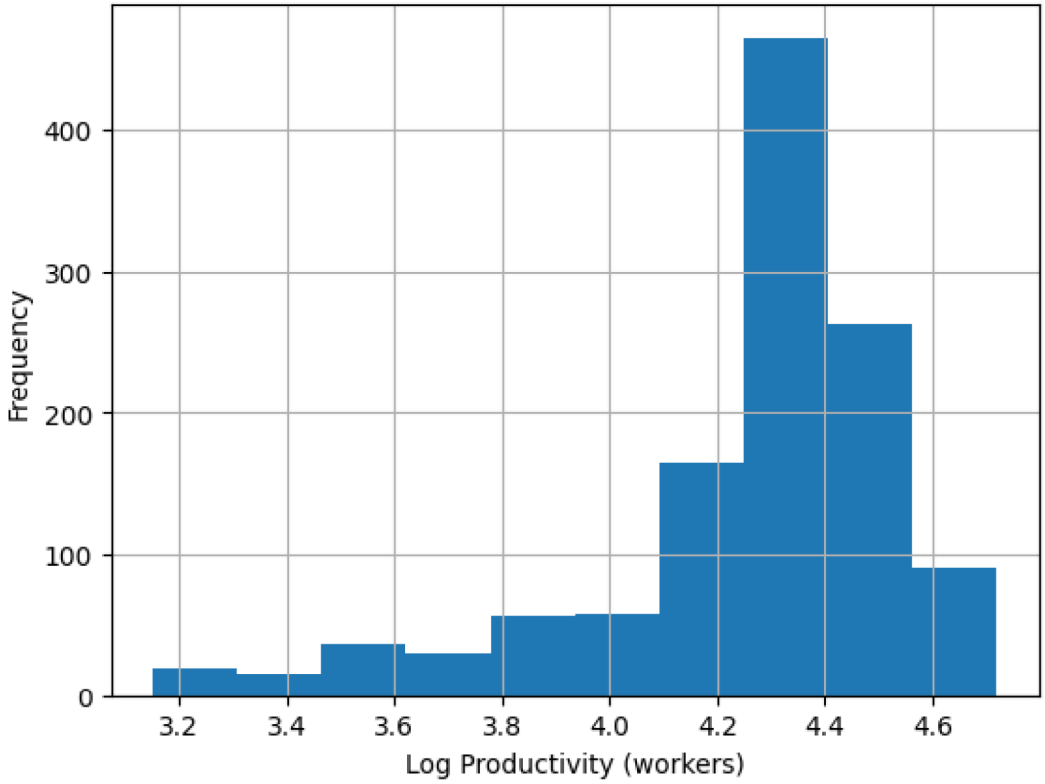
[ ]:
```
import matplotlib.pyplot as plt
d["actual_productivity"].hist(bins=10)
plt.xlabel('Actual Productivity (workers)')
plt.ylabel('Frequency')
plt.show()
```



[ ]:
```
d["actual_productivity"].describe()
```

[ ]:
```
count    1197.000000
mean        0.735091
std         0.174488
min         0.233705
25%         0.650307
50%         0.773333
75%         0.850253
max         1.120437
Name: actual_productivity, dtype: float64
```

```
[ ]: d["log_productivity"].hist(bins=10)
     plt.xlabel('Log Productivity (workers)')
     plt.ylabel('Frequency')
     plt.show()
```




```
[ ]: d["log_productivity"].describe()
```

```
[ ]: count    1197.000000
     mean        4.261224
     std         0.289384
     min         3.151477
     25%         4.174860
     50%         4.348125
     75%         4.442948
     max         4.718889
     Name: log_productivity, dtype: float64
```

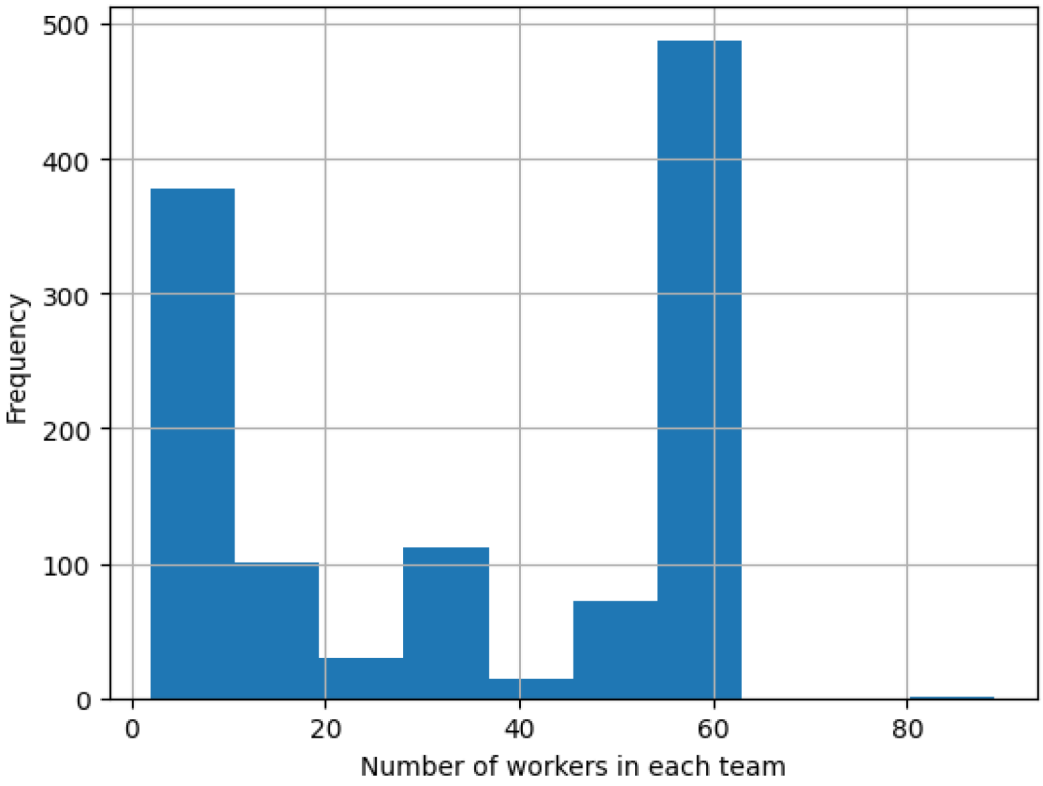
Workers:

The number of workers values decreased when transformed into log_no_of_workers, with the range initially spanning from about 2 to 89, and then narrowing to approximately 0.69 to 4.49. This narrower range indicates less dispersion in the data. Additionally, there is a higher peak in

the log_no_of_workers distribution, suggesting that a significant number of observations cluster within a particular range on the logarithmic scale.

Therefore, the transformation into log_no_of_workers leads to a distribution that more closely resembles a normal distribution compared to its original scale.

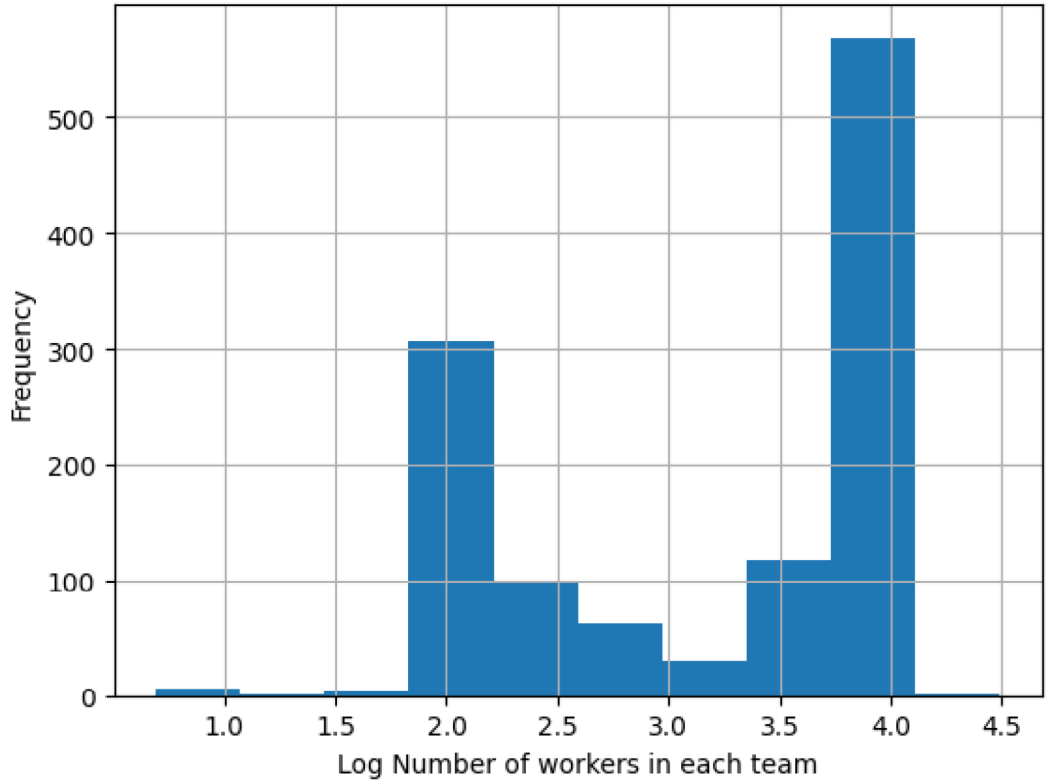
[ ]:
```
d["no_of_workers"].hist(bins=10)
plt.xlabel('Number of workers in each team')
plt.ylabel('Frequency')
plt.show()
```



[ ]:
```
d["no_of_workers"].describe()
```

[ ]:
```
count    1197.000000
mean       34.609858
std        22.197687
min         2.000000
25%         9.000000
50%        34.000000
75%        57.000000
max        89.000000
Name: no_of_workers, dtype: float64
```

```
[ ]: d["log_no_of_workers"].hist(bins=10)
     plt.xlabel('Log Number of workers in each team')
     plt.ylabel('Frequency')
     plt.show()
```




```
[ ]: d["log_no_of_workers"].describe()
```

```
[ ]: count    1197.000000
     mean        3.231103
     std         0.874730
     min         0.693147
     25%         2.197225
     50%         3.526361
     75%         4.043051
     max         4.488636
     Name: log_no_of_workers, dtype: float64
```

b. Each month is divided into five quarters, where approximately each week is a quarter. How does the distribution of logarithm of productivity change in each quarter?

1. The number of observations (count) varies across quarters, with Quarter1 having the highest count (360) and Quarter5 having the lowest count (44).

2. The mean and median (50th percentile) of the logarithm of productivity are relatively consistent across quarters, with Quarter5 having slightly higher values compared to the other quarters.
3. The standard deviation varies slightly across quarters, indicating differences in the dispersion of the logarithm of productivity.
4. The minimum and maximum values show the range of logarithm of productivity within each quarter, with Quarter5 having the highest maximum value.

[ ]:
```
log_productivity_quarter = d.groupby('quarter')['log_productivity']
log_productivity_quarter.describe()
```

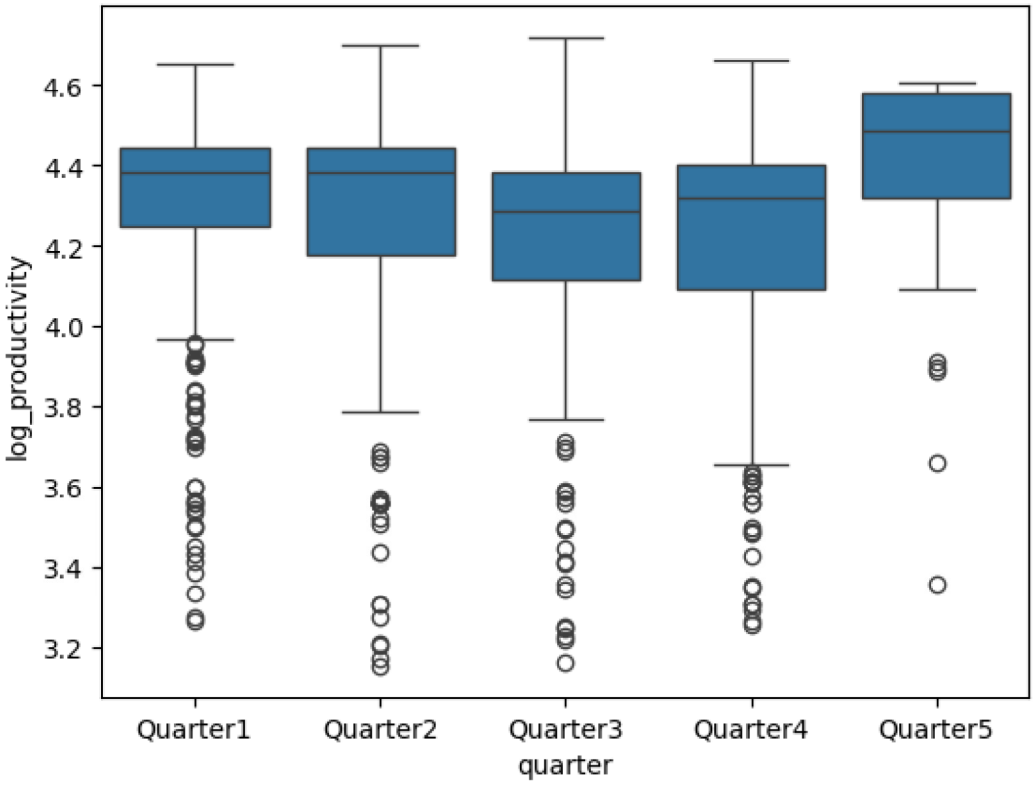
[ ]:

quarter	count	mean	std	min	25%	50%	75%	max
Quarter1	360.0	4.290405	0.259116	3.261854	4.248446	4.382083	4.442792	4.654595
Quarter2	335.0	4.274406	0.285213	3.151477	4.177887	4.382070	4.443242	4.700920
Quarter3	210.0	4.215843	0.302868	3.160380	4.117134	4.284180	4.382515	4.718889
Quarter4	248.0	4.218121	0.314575	3.255690	4.095062	4.318036	4.399883	4.663082
Quarter5	44.0	4.381652	0.280099	3.356843	4.318350	4.486625	4.576634	4.605628

Create a box plot of logarithm of productivity by quarter. Comment on your observations. Does the worker productivity increase towards the end of the month (quarter 5) as compared to other quarters?

Quarter 5 appears to have a slightly higher median and upper quartile compared to the other quarters, suggesting potentially higher productivity towards the end of the month.

[ ]:
```
import seaborn as sns
sns.boxplot(x='quarter', y='log_productivity', data=d)
plt.show()
```



Perform a t-test for quarter 5 with respect to (individually) all other quarters. (Hint. There will be 4 different t-tests). What do you observe for each t-test?

Quarter 5 displayed a statistically significant increase in productivity compared to Quarter 1 ($p = 0.029$), Quarter 2 ($p = 0.019$), Quarter 3 ($p = 0.001$), and Quarter 4 ($p = 0.001$).

[ ]:
```
import pandas as pd
from scipy.stats import ttest_ind

quarter_5_data = d[d['quarter'] == 'Quarter5']
other_quarters_data = d[d['quarter'] != 'Quarter5']

for quarter, data in other_quarters_data.groupby('quarter'):
    t_stat, p_value = ttest_ind(quarter_5_data['log_productivity'],␣
 ↪data['log_productivity'])
    print(f"T-test for Quarter 5 vs {quarter}:")
    print(f"T-statistic: {t_stat}, p-value: {p_value}")
    if p_value < 0.05:
        print("Reject null hypothesis: There is a significant difference.")
    else:
        print("Fail to reject null hypothesis: There is no significant␣
 ↪difference.")
```

```
    print()
```

```
T-test for Quarter 5 vs Quarter1:
T-statistic: 2.185396025567071, p-value: 0.02943633151153009
Reject null hypothesis: There is a significant difference.

T-test for Quarter 5 vs Quarter2:
T-statistic: 2.349743311663638, p-value: 0.019301427713836954
Reject null hypothesis: There is a significant difference.

T-test for Quarter 5 vs Quarter3:
T-statistic: 3.343507254044234, p-value: 0.000953075291086163
Reject null hypothesis: There is a significant difference.

T-test for Quarter 5 vs Quarter4:
T-statistic: 3.2278344431012584, p-value: 0.0013903605543610135
Reject null hypothesis: There is a significant difference.

T-test for Quarter 5 vs Quarter5:
T-statistic: nan, p-value: nan
Fail to reject null hypothesis: There is no significant difference.
```

Comment on the findings. Use a 95% confidence. (You need to state the hypotheses explicitly in your answer, the mean and standard deviations for each of the groups in a t-tests, the t-statistics and the p-values. Then you need to explain what the p-value means.).

Hypotheses:

Null Hypothesis (H0): There is no difference in productivity between Quarter 5 and each of the other quarters.

Alternative Hypothesis (H1): There is a difference in productivity between Quarter 5 and each of the other quarters.

With a 95% confidence level, the p-values for all t-tests are less than 0.05, indicating that we reject the null hypothesis for each comparison. This means there is sufficient evidence to suggest that Quarter 5 has significantly higher productivity compared to each of the other quarters.

p-value: The p-value in hypothesis testing measures the probability of observing the data if the null hypothesis (no effect or no difference) is true. A low p-value (typically below 0.05) indicates strong evidence against the null hypothesis, suggesting that the observed differences are unlikely to be due to chance.

[ ]:
```
log_productivity_quarter = d.groupby('quarter')['log_productivity']
log_productivity_quarter.agg([np.mean, np.std])
```

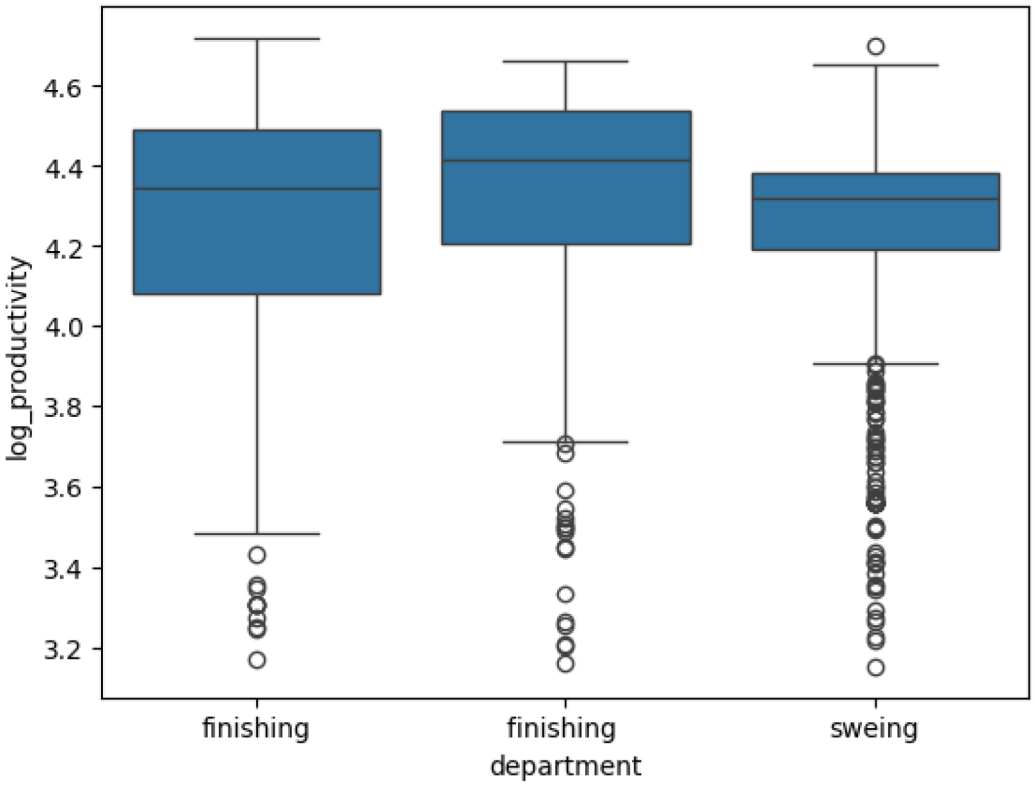
[ ]:
```
              mean       std
quarter
```

```
Quarter1  4.290405  0.259116
Quarter2  4.274406  0.285213
Quarter3  4.215843  0.302868
Quarter4  4.218121  0.314575
Quarter5  4.381652  0.280099
```

c. Repeat part (b) for department instead of quarter, day instead of quarter, and no_of_style_change instead of quarter. In these cases, perform the t-test for all pairs of departments and all pairs of style changes. For day, compare Sunday with all other weekdays.

Department

[ ]:
```
sns.boxplot(x='department', y='log_productivity', data=d)
plt.show()
```



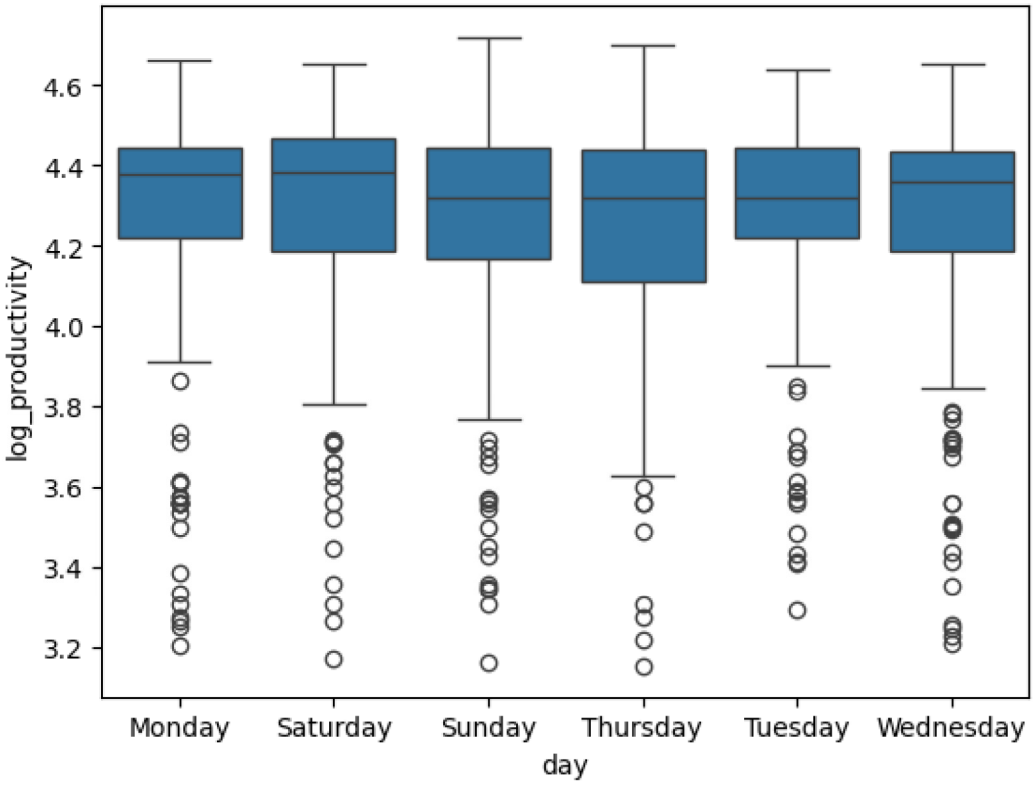
[ ]:
```
departments = d['department'].unique()

for i in range(len(departments)):
    for j in range(i+1, len(departments)):
        dept1_data = d[d['department'] == departments[i]]['log_productivity']
        dept2_data = d[d['department'] == departments[j]]['log_productivity']
```

```
        t_statistic, p_value = ttest_ind(dept1_data, dept2_data)

        print(f"T-test for {departments[i]} vs {departments[j]}:")
        print(f"T-statistic: {t_statistic}, p-value: {p_value}")
        if p_value < 0.05:
            print("Reject null hypothesis: There is a significant difference.")
        else:
            print("Fail to reject null hypothesis: There is no significant␣
↪difference.")
        print()
```

```
T-test for sweing vs finishing :
T-statistic: -3.552595110520864, p-value: 0.00040028431437478574
Reject null hypothesis: There is a significant difference.

T-test for sweing vs finishing:
T-statistic: 0.9268299378480578, p-value: 0.3542532447620479
Fail to reject null hypothesis: There is no significant difference.

T-test for finishing  vs finishing:
T-statistic: 3.1565021267111875, p-value: 0.0016922963741435742
Reject null hypothesis: There is a significant difference.
```

Day

[ ]:
```
sns.boxplot(x='day', y='log_productivity', data=d)
plt.show()
```



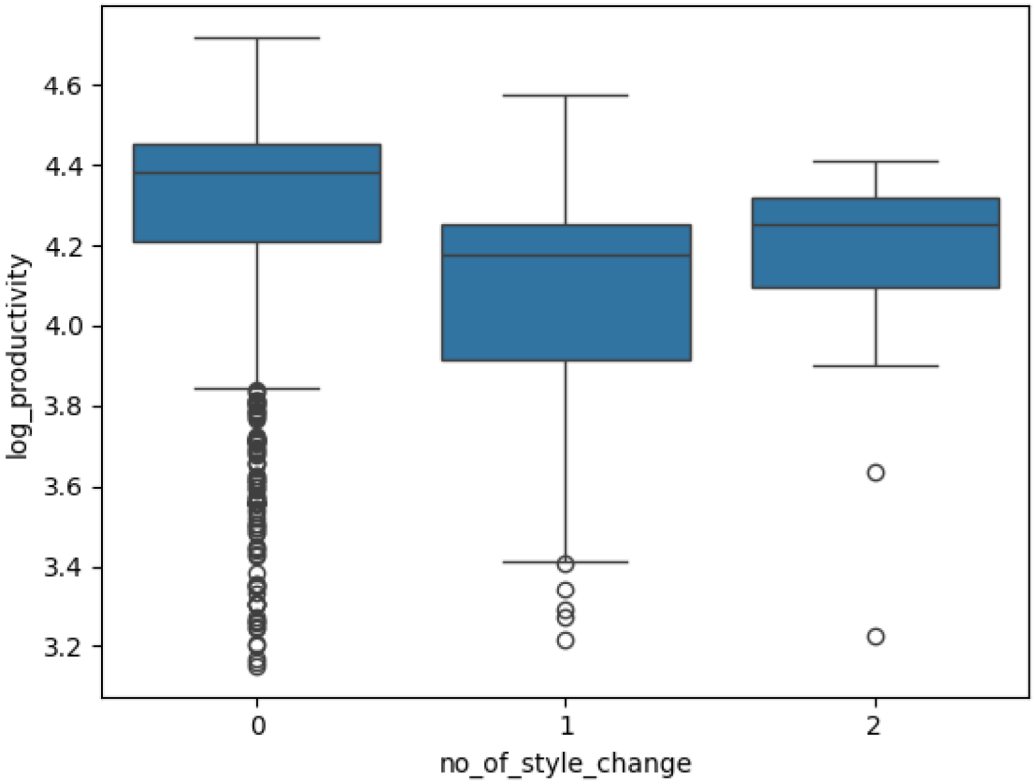
[ ]:
```
weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']

for weekday in weekdays:
    sunday_data = d[d['day'] == 'Sunday']['log_productivity']
    weekday_data = d[d['day'] == weekday]['log_productivity']

    t_statistic, p_value = ttest_ind(sunday_data, weekday_data)

    print(f"T-test for Sunday vs {weekday}:")
    print(f"T-statistic: {t_statistic}, p-value: {p_value}")
    if p_value < 0.05:
        print("Reject null hypothesis: There is a significant difference.")
    else:
        print("Fail to reject null hypothesis: There is no significant␣
 ↪difference.")
    print()
```

```
T-test for Sunday vs Monday:
T-statistic: -0.2508138187485118, p-value: 0.8020868418718979
Fail to reject null hypothesis: There is no significant difference.
```

```
T-test for Sunday vs Tuesday:
T-statistic: -0.9561959422992043, p-value: 0.3395479101161385
Fail to reject null hypothesis: There is no significant difference.

T-test for Sunday vs Wednesday:
T-statistic: -0.0300827736223543, p-value: 0.9760157128073305
Fail to reject null hypothesis: There is no significant difference.

T-test for Sunday vs Thursday:
T-statistic: 0.2088609242357079, p-value: 0.8346631275525696
Fail to reject null hypothesis: There is no significant difference.

T-test for Sunday vs Friday:
T-statistic: nan, p-value: nan
Fail to reject null hypothesis: There is no significant difference.

T-test for Sunday vs Saturday:
T-statistic: -1.1534970911823332, p-value: 0.24941647550102006
Fail to reject null hypothesis: There is no significant difference.
```

Number of style change

[ ]:
```
sns.boxplot(x='no_of_style_change', y='log_productivity', data=d)
plt.show()
```



[ ]:
```
style_changes = d['no_of_style_change'].unique()

for i in range(len(style_changes)):
    for j in range(i+1, len(style_changes)):
        style_change1_data = d[d['no_of_style_change'] ==␣
↪style_changes[i]]['log_productivity']
        style_change2_data = d[d['no_of_style_change'] ==␣
↪style_changes[j]]['log_productivity']

        t_statistic, p_value = ttest_ind(style_change1_data, style_change2_data)

        print(f"T-test for Style Change {style_changes[i]} vs Style Change␣
↪{style_changes[j]}:")
        print(f"T-statistic: {t_statistic}, p-value: {p_value}")
        if p_value < 0.05:
            print("Reject null hypothesis: There is a significant difference.")
        else:
            print("Fail to reject null hypothesis: There is no significant␣
↪difference.")
        print()
```

```
T-test for Style Change 0 vs Style Change 1:
T-statistic: 7.317166413470184, p-value: 4.711568034626225e-13
Reject null hypothesis: There is a significant difference.

T-test for Style Change 0 vs Style Change 2:
T-statistic: 2.3215444234359723, p-value: 0.02044265784688274
Reject null hypothesis: There is a significant difference.

T-test for Style Change 1 vs Style Change 2:
T-statistic: -1.566692700351746, p-value: 0.11936631668898746
Fail to reject null hypothesis: There is no significant difference.
```

d. Perform a scatter plot of the natural logarithm of no_of_workers +1 on x–axis and natural logarithm of productivity on y-axis. What do you observe? Comment on any pattern that you may observe. Report the correlation coefficient between the two variables.

[ ]:
```
d["log_no_of_workers_1"]= np.log(d['no_of_workers'] + 1)
d.head()
```

[ ]:

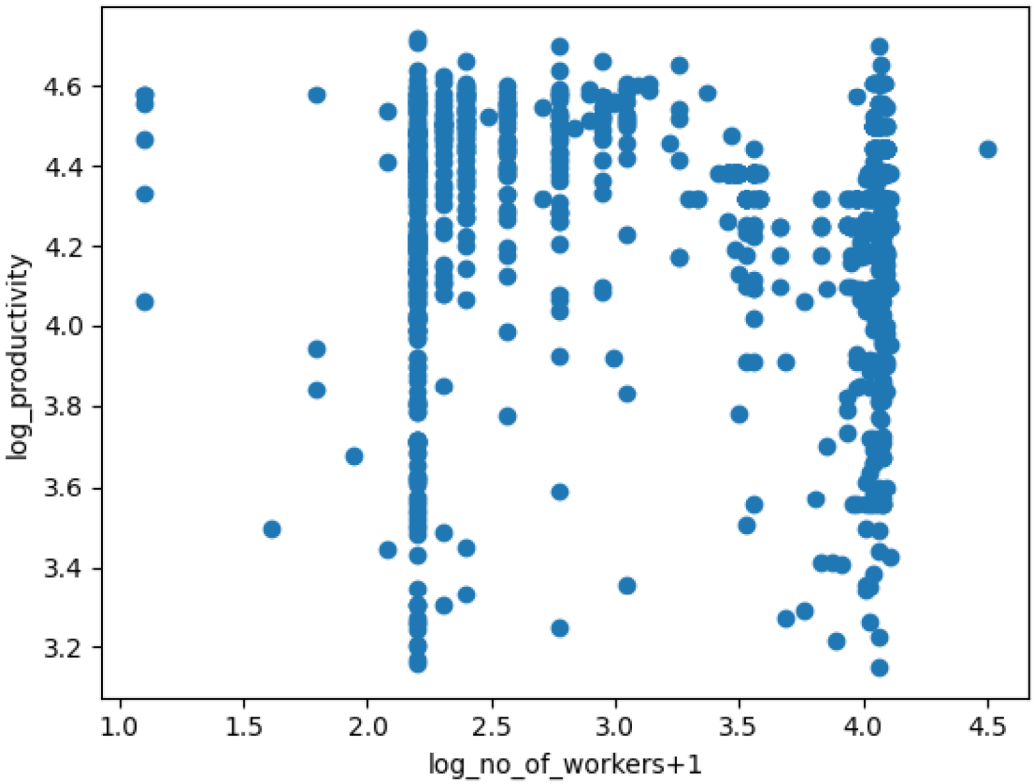
	date	quarter	department	day	team	targeted_productivity	\
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	
2	1/1/2015	Quarter1	sweing	Thursday	11	0.80	
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	
4	1/1/2015	Quarter1	sweing	Thursday	6	0.80	

	smv	over_time	incentive	idle_time	idle_men	no_of_style_change	\
0	26.16	7080	98	0.0	0	0	
1	3.94	960	0	0.0	0	0	
2	11.41	3660	50	0.0	0	0	
3	11.41	3660	50	0.0	0	0	
4	25.90	1920	50	0.0	0	0	

	no_of_workers	actual_productivity	log_productivity	log_no_of_workers	\
0	59.0	0.940725	4.544066	4.077537	
1	8.0	0.886500	4.484696	2.079442	
2	30.5	0.800570	4.382739	3.417727	
3	30.5	0.800570	4.382739	3.417727	
4	56.0	0.800382	4.382504	4.025352	

	log_no_of_workers_1
0	4.094345
1	2.197225
2	3.449988
3	3.449988
4	4.043051

[ ]:
```
plt.scatter(x=d['log_no_of_workers_1'], y=d['log_productivity'])
plt.xlabel('log_no_of_workers+1')
plt.ylabel('log_productivity')
plt.show()
```



The correlation coefficient between the variables log_no_of_workers_1 and log_productivity is approximately -0.00286. The change in the number of workers are not significantly associated with changes in productivity, based on the linear correlation.

[ ]:
```
correlation_coefficient = d['log_no_of_workers_1'].corr(d['log_productivity'])
print(correlation_coefficient)
```

```
-0.002860469596148367
```

e. Perform a scatter plot of the natural logarithm of incentive + 1 on x-axis and natural logarithm of productivity on y-axis. What do you observe? Comment on any patterns that you may observe. Report the correlation coefficient between the two variables.

[ ]:
```
d["log_incentive_1"]= np.log(d['incentive'] + 1)
d.head()
```

[ ]:

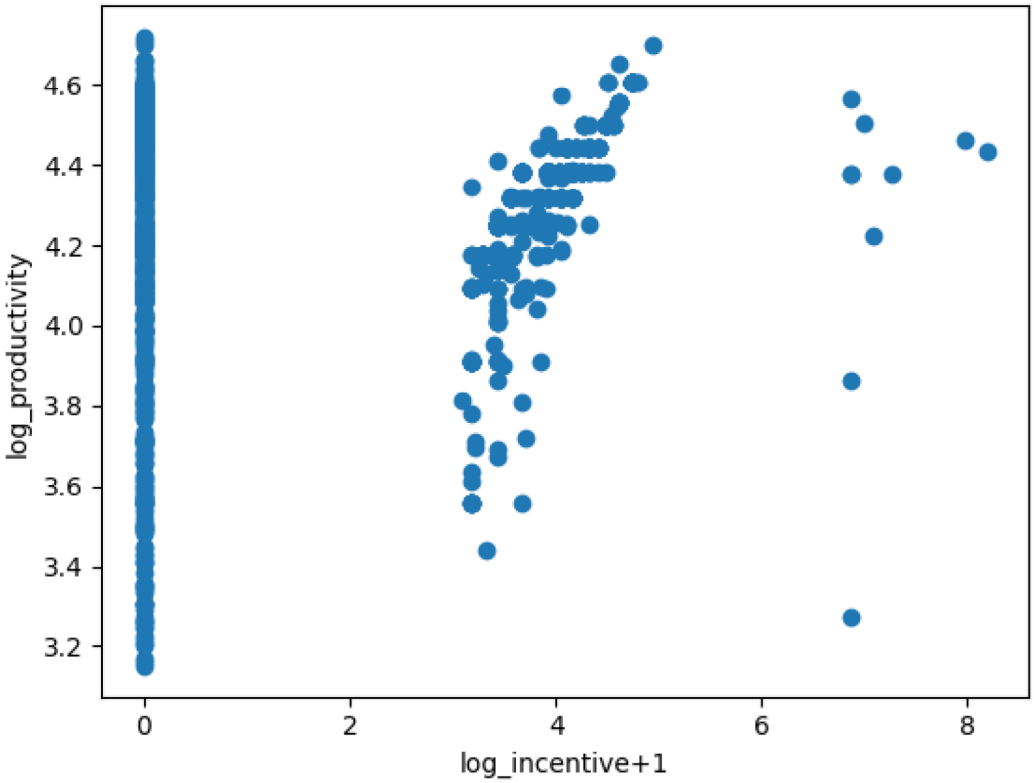
	date	quarter	department	day	team	targeted_productivity	\
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	
2	1/1/2015	Quarter1	sweing	Thursday	11	0.80	
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	
4	1/1/2015	Quarter1	sweing	Thursday	6	0.80	

	smv	over_time	incentive	idle_time	idle_men	no_of_style_change	\
0	26.16	7080	98	0.0	0	0	
1	3.94	960	0	0.0	0	0	
2	11.41	3660	50	0.0	0	0	
3	11.41	3660	50	0.0	0	0	
4	25.90	1920	50	0.0	0	0	

	no_of_workers	actual_productivity	log_productivity	log_no_of_workers	\
0	59.0	0.940725	4.544066	4.077537	
1	8.0	0.886500	4.484696	2.079442	
2	30.5	0.800570	4.382739	3.417727	
3	30.5	0.800570	4.382739	3.417727	
4	56.0	0.800382	4.382504	4.025352	

	log_no_of_workers_1	log_incentive_1
0	4.094345	4.595120
1	2.197225	0.000000
2	3.449988	3.931826
3	3.449988	3.931826
4	4.043051	3.931826

[ ]:
```
plt.scatter(x=d['log_incentive_1'], y=d['log_productivity'])
plt.xlabel('log_incentive+1')
plt.ylabel('log_productivity')
plt.show()
```



The correlation coefficient between the variables log_incentive_1 and log_productivity is approximately 0.217063. This correlation is not very strong, but it is still worth noting.

[ ]:
```
correlation_coefficient_1 = d['log_incentive_1'].
  ↪corr(d['log_productivity'],method='spearman')
print(correlation_coefficient_1)
```

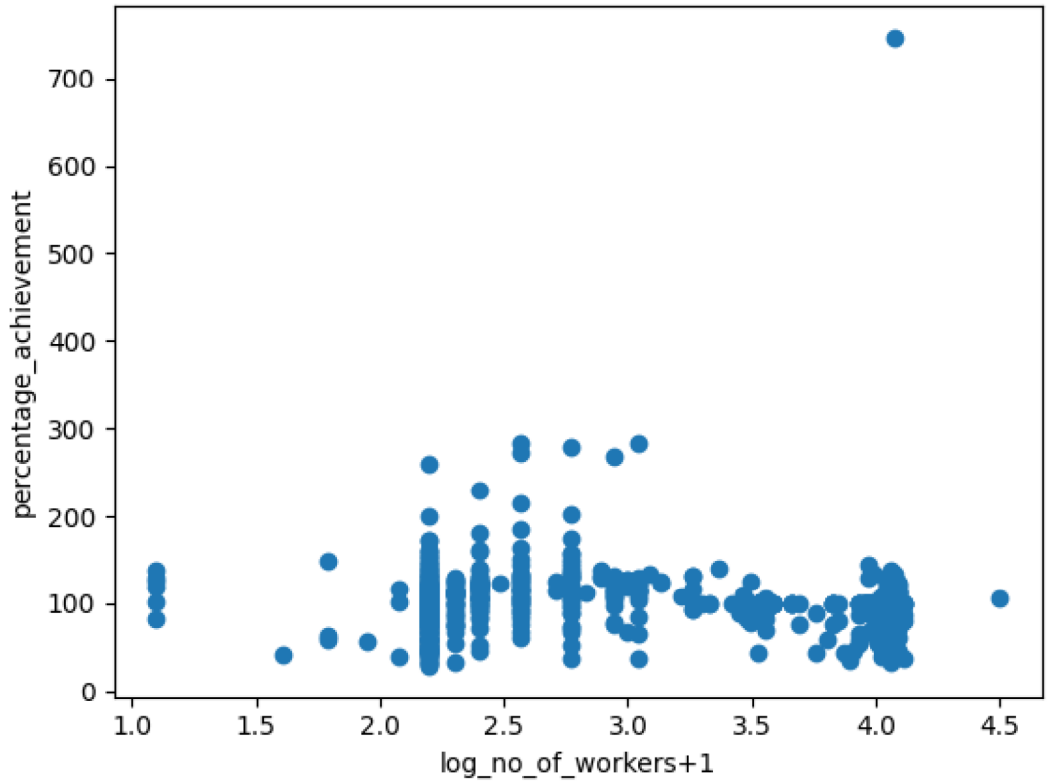
```
0.21706358696654462
```

f. Repeat (d) and (e) for percentage_achievement instead of logarithm of productivity.

[ ]:
```
#Percentage Achievement=( Targeted Productivity/ Actual Productivity)×100%
d['percentage_achievement'] = (d['actual_productivity'] /␣
  ↪d['targeted_productivity']) * 100
d["percentage_achievement"].head()
```

[ ]:
```
0    117.590678
1    118.200000
2    100.071311
3    100.071311
4    100.047743
Name: percentage_achievement, dtype: float64
```

[ ]:
```
# number of workers and percentage achievement
plt.scatter(x=d['log_no_of_workers_1'], y=d['percentage_achievement'])
plt.xlabel('log_no_of_workers+1')
plt.ylabel('percentage_achievement')
plt.show()
```

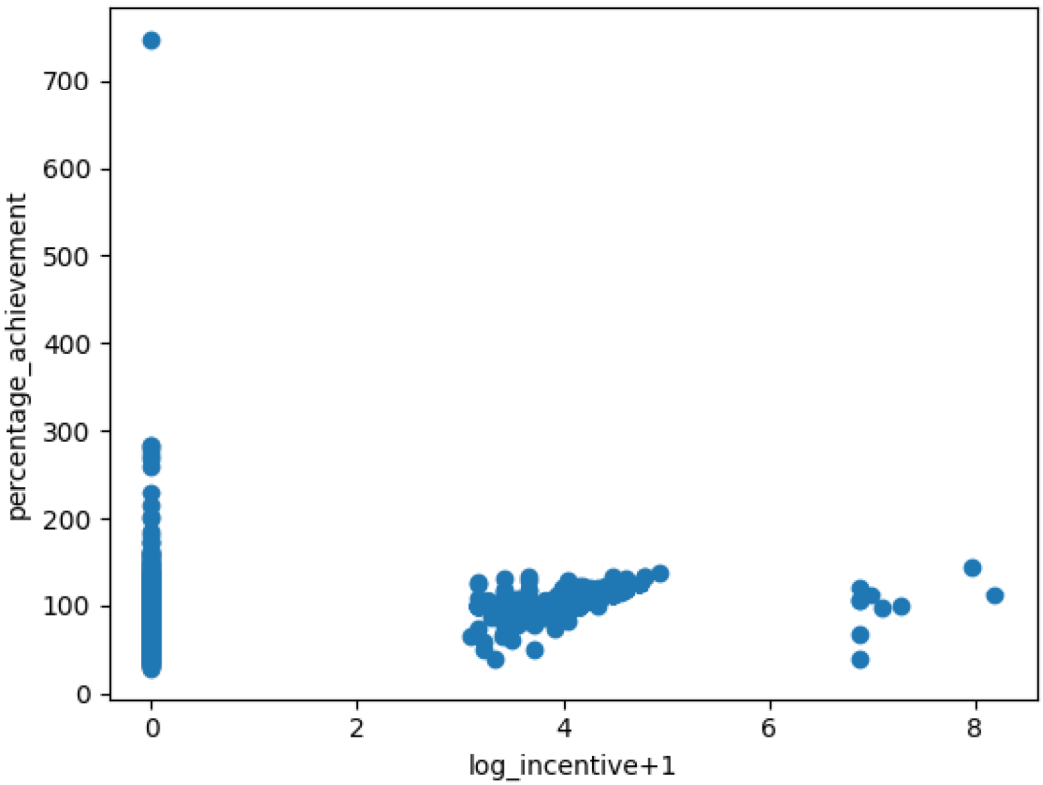


The correlation coefficient between the variables log_no_of_workers_1 and percentage_achievement is approximately-0.00802. The change in the number of workers are not significantly associated with hanges in productivity, based on the linear correlation.

[ ]:
```
correlation_coefficient = d['log_no_of_workers_1'].
  ↪corr(d['percentage_achievement'])
print(correlation_coefficient)
```

```
-0.008026734383457825
```

[ ]:
```
# number of workers and percentage achievement
plt.scatter(x=d['log_incentive_1'], y=d['percentage_achievement'])
plt.xlabel('log_incentive+1')
plt.ylabel('percentage_achievement')
plt.show()
```



The correlation coefficient between the variables log_incentive_1 and percentage_achievement is approximately 0.136517. This correlation is not very strong, but it is still worth noting.

[ ]:
```
correlation_coefficient_1 = d['log_incentive_1'].
  ↪corr(d['percentage_achievement'],method='spearman')
print(correlation_coefficient_1)
```

```
0.1365174303878588
```

3. Regression Analysis – 40 points.

a. Estimate an ordinary least square repression (OLS) with natural logarithm of productivity as response variable and natural logarithm of no_of_workers + 1 as the predictor variable.Comment on the relationship between the response and the predictor variable.

[ ]:
```
import statsmodels.api as sm
X = d['log_no_of_workers_1']
y = d['log_productivity']
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       log_productivity   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                 -0.001
Method:                 Least Squares   F-statistic:                  0.009778
Date:                Fri, 01 Mar 2024   Prob (F-statistic):              0.921
Time:                        09:50:29   Log-Likelihood:                -213.68
No. Observations:                1197   AIC:                             431.4
Df Residuals:                    1195   BIC:                             441.5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=======
                           coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const                    4.2645      0.034    124.471      0.000       4.197
4.332
log_no_of_workers_1     -0.0010      0.010     -0.099      0.921      -0.021
0.019
==============================================================================
Omnibus:                      341.342   Durbin-Watson:                   0.823
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              780.960
Skew:                          -1.574   Prob(JB):                    2.61e-170
Kurtosis:                       5.398   Cond. No.                         15.0
==============================================================================

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

Is team size (number of workers in a team) a good predictor of productivity?

The R-squared value is 0.000, indicating that the model explains none of the variance in the response variable, log_productivity.

Does this finding conform to the exploratory analysis in 2(d)?

Yes, it conform to 2(d), The change in the number of workers are not significantly associated with changes in productivity.

What is the estimated regression equation?

Estimated regression equation:

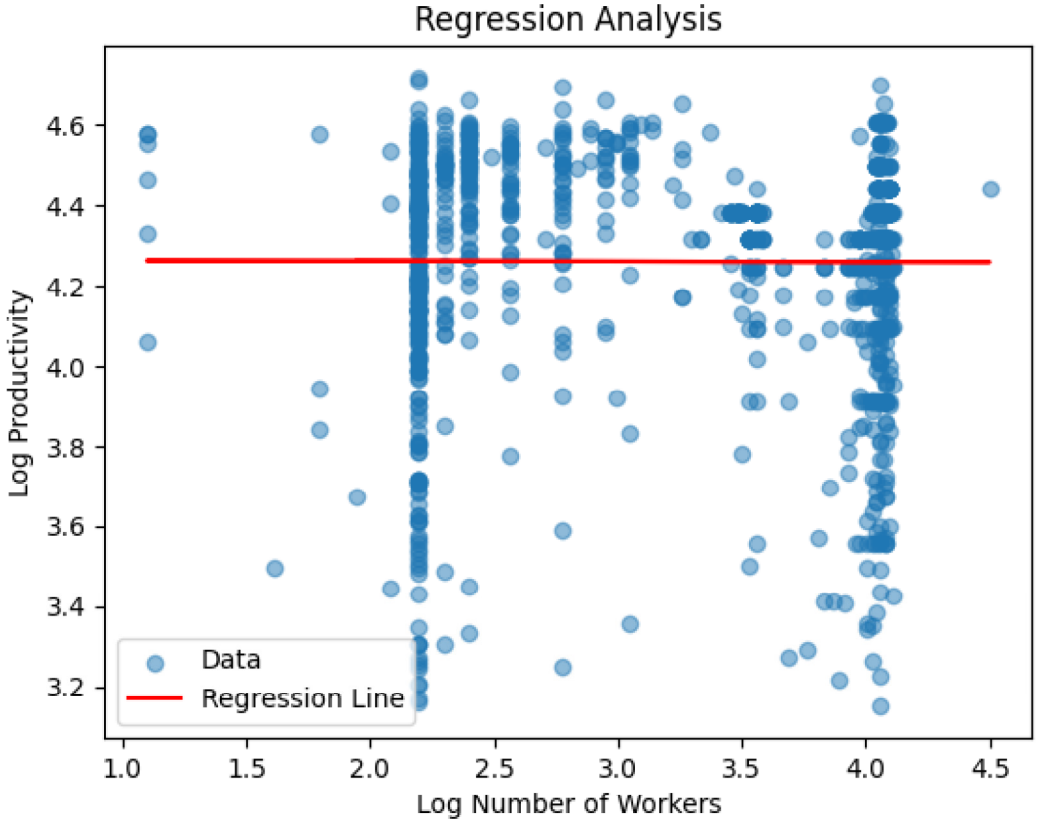
log_productivity= 4.2645 - 0.0010(log_no_of_workers_1)

How much of the variance in the response is explained by the predictor? (Comment on the R-square, the intercept, slope, and the t-statistics of the intercept and slope, and the p-values).

The regression model indicates a very low explanatory power, with an R-squared value of 0.0, suggesting that the predictor variable, log_no_of_workers_1, does not explain any variance in the productivity. Both the intercept and slope are not statistically significant ($p > 0.05$), indicating that they are not reliable predictors of productivity.

Finally, plot the regression equation on the scatterplot of the predictor and response.

[ ]:
```
import matplotlib.pyplot as plt
plt.scatter(d['log_no_of_workers_1'], d['log_productivity'], alpha=0.5,␣
 ↪label='Data')
plt.plot(d['log_no_of_workers_1'], model.predict(sm.
 ↪add_constant(d['log_no_of_workers_1'])), color='red', label='Regression␣
 ↪Line')
plt.xlabel('Log Number of Workers')
plt.ylabel('Log Productivity')
plt.title('Regression Analysis')
plt.legend()
plt.show()
```



b. Repeat (a) with logarithm of incentives + 1 as the predictor.

[ ]:
```
X = d['log_incentive_1']
y = d['log_productivity']
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
predictions = model.predict(X)
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       log_productivity   R-squared:                       0.046
Model:                            OLS   Adj. R-squared:                  0.045
Method:                 Least Squares   F-statistic:                     57.91
Date:                Fri, 01 Mar 2024   Prob (F-statistic):           5.54e-14
Time:                        09:50:29   Log-Likelihood:                -185.37
No. Observations:                1197   AIC:                             374.7
Df Residuals:                    1195   BIC:                             384.9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
const          4.2009      0.011    368.832      0.000       4.179
4.223
log_incentive_1     0.0307      0.004      7.610      0.000       0.023
0.039
==============================================================================
Omnibus:                      273.631   Durbin-Watson:                   0.896
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              550.579
Skew:                          -1.320   Prob(JB):                    2.77e-120
Kurtosis:                       5.019   Cond. No.                         4.18
==============================================================================

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

Is woker's incentive a good predictor of productivity?

The R-squared value is 0.046, which means that approximately 4.6% of the variance in log_productivity is explained by the predictor variable log_incentive_1.

Does this finding conform to the exploratory analysis in 2(e)?

Yes, it conform to 2(e), The change in the workers' incentive are slightly associated with changes in productivity.

What is the estimated regression equation?

estimated regression equation:

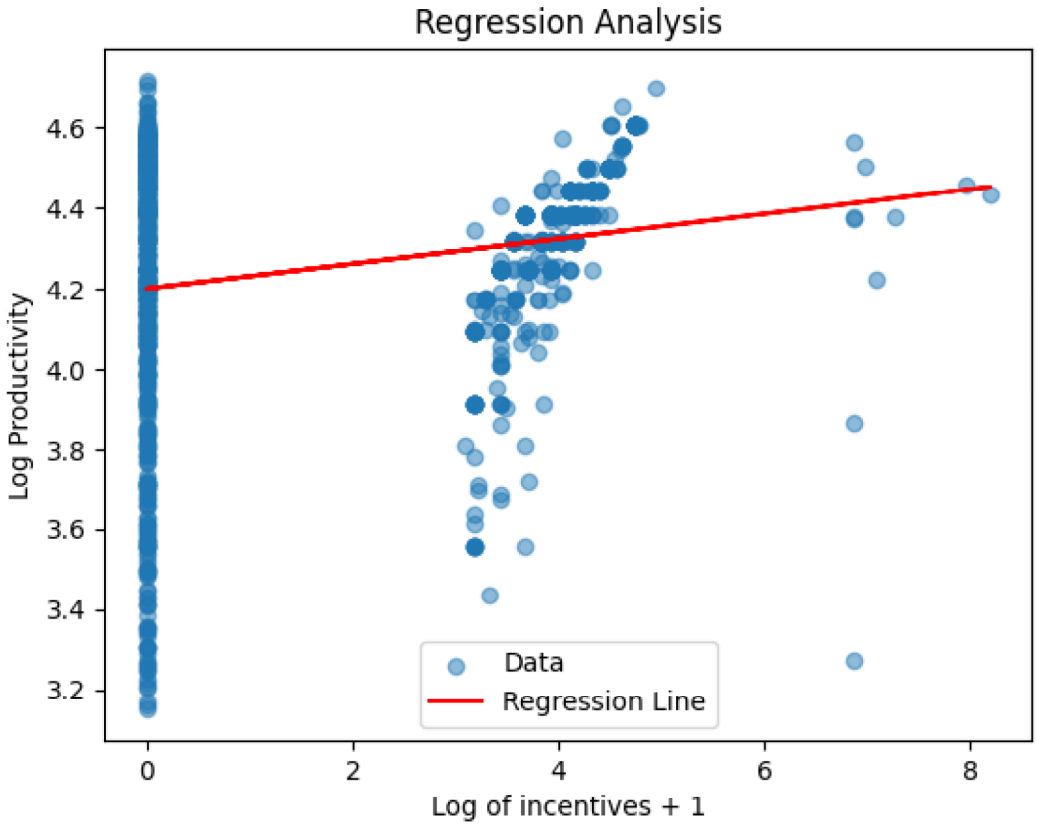
log_productivity = 4.2009 + 0.0307(log_incentive_1)

How much of the variance in the response is explained by the predictor? (Comment on the R-square, the intercept, slope, and the t-statistics of the intercept and slope, and the p-values).

The R-squared value is 0.046, indicating that approximately 4.6% of the variance in log_productivity is explained by the predictor variable log_incentive_1. The t-statistic for the intercept is very high (368.832) with p-values close to zero, indicating that the intercept is statistically significant.

plot the regression equation on the scatterplot of the predictor and response.

[ ]:
```
import matplotlib.pyplot as plt
plt.scatter(d['log_incentive_1'], d['log_productivity'], alpha=0.5,␣
 ↪label='Data')
plt.plot(d['log_incentive_1'], model.predict(sm.
 ↪add_constant(d['log_incentive_1'])), color='red', label='Regression Line')
plt.xlabel('Log of incentives + 1')
plt.ylabel('Log Productivity')
plt.title('Regression Analysis')
plt.legend()
plt.show()
```



c. Estimate the regression equation for log of actual productivity as response and the following variables as predictors: log of no_of_workers + 1, log of incentive + 1, log of targeted productivity, no_of_style_change, quarter (factor variable), department (factor variable), day (factor variable) and team (factor variable). Show the regression summary.

[ ]:
```
d["log_targeted_productivity"] = np.log(d['targeted_productivity'])
d["log_actual_productivity"] = np.log(d['actual_productivity'])
```

[ ]:
```
d['const'] = 1
X = d[['const', 'log_no_of_workers_1', 'log_incentive_1',␣
 ↪'log_targeted_productivity', 'no_of_style_change', 'quarter', 'department',␣
 ↪'day', 'team']]
y = d['log_actual_productivity']
X = pd.get_dummies(X, columns=['quarter', 'department', 'day', 'team'],␣
 ↪drop_first=True)
model = sm.OLS(y, X).fit()
print(model.summary())
```

OLS Regression Results

```
==============================================================================
===
Dep. Variable:     log_actual_productivity   R-squared:
0.325
Model:                                 OLS   Adj. R-squared:
0.310
Method:                      Least Squares   F-statistic:
21.64
Date:                     Fri, 01 Mar 2024   Prob (F-statistic):
9.80e-82
Time:                             09:59:20   Log-Likelihood:
21.290
No. Observations:                     1197   AIC:
11.42
Df Residuals:                         1170   BIC:
148.8
Df Model:                               26
Covariance Type:                 nonrobust
==============================================================================
=============
                                 coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-------------
const                         -0.6312      0.085     -7.463      0.000
-0.797      -0.465
log_no_of_workers_1            0.1902      0.031      6.069      0.000
0.129       0.252
log_incentive_1                0.0672      0.006     11.401      0.000
0.056       0.079
log_targeted_productivity      0.4789      0.043     11.166      0.000
0.395       0.563
no_of_style_change            -0.0276      0.019     -1.474      0.141
-0.064       0.009
quarter_Quarter2              -0.0028      0.019     -0.149      0.882
-0.039       0.034
quarter_Quarter3              -0.0365      0.021     -1.716      0.086
-0.078       0.005
quarter_Quarter4              -0.0531      0.021     -2.564      0.010
-0.094      -0.012
quarter_Quarter5               0.0817      0.040      2.027      0.043
0.003       0.161
department_finishing           0.1085      0.022      4.936      0.000
0.065       0.152
department_sweing             -0.4663      0.056     -8.333      0.000
-0.576      -0.357
day_Saturday                   0.0303      0.025      1.205      0.228
-0.019       0.080
```

```
day_Sunday                    0.0171      0.024      0.706      0.480
-0.030       0.065
day_Thursday                  0.0171      0.025      0.696      0.487
-0.031       0.065
day_Tuesday                   0.0413      0.024      1.713      0.087
-0.006       0.089
day_Wednesday                 0.0244      0.024      1.019      0.309
-0.023       0.071
team_2                       -0.0469      0.033     -1.418      0.157
-0.112       0.018
team_3                       -0.0128      0.034     -0.372      0.710
-0.080       0.055
team_4                       -0.0405      0.034     -1.204      0.229
-0.106       0.025
team_5                       -0.0604      0.035     -1.733      0.083
-0.129       0.008
team_6                       -0.0770      0.036     -2.144      0.032
-0.147      -0.007
team_7                       -0.1274      0.034     -3.693      0.000
-0.195      -0.060
team_8                       -0.1212      0.033     -3.635      0.000
-0.187      -0.056
team_9                       -0.1022      0.033     -3.063      0.002
-0.168      -0.037
team_10                      -0.1235      0.034     -3.660      0.000
-0.190      -0.057
team_11                      -0.1370      0.035     -3.904      0.000
-0.206      -0.068
team_12                       0.0062      0.035      0.176      0.860
-0.063       0.075
==============================================================================
Omnibus:                      353.196   Durbin-Watson:                   1.220
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1272.479
Skew:                          -1.402   Prob(JB):                    4.84e-277
Kurtosis:                       7.202   Cond. No.                         67.5
==============================================================================

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

i. Which of the following variables significantly affect worker productivity and which direction? State the level of significance.

- Log_no_of_workers_1: A one-unit increase in log_no_of_workers_1 is associated with a 0.1902 increase in log_actual_productivity ($p < 0.05$), indicating a positive relationship between the number of workers and productivity.
- Log_incentive_1: A one-unit increase in log_incentive_1 results in a 0.0672 increase in log_actual_productivity ($p < 0.05$), suggesting that higher incentives are linked to higher

productivity.

- Log_targeted_productivity: A one-unit increase in log_targeted_productivity corresponds to a 0.4789 increase in log_actual_productivity ($p < 0.05$), indicating a positive relationship between targeted productivity and actual productivity.
- Both department_finishing and department_sewing significantly influence productivity. The coefficient for department_finishing is 0.1085 ($p < 0.05$), indicating higher productivity in the finishing department compared to the reference department. Conversely, the coefficient for department_sewing is -0.4663 ($p < 0.05$), indicating lower productivity in the sewing department compared to the reference department.

Other variables such as the day of the week and team assignments do not significantly affect productivity as their p-values are above the threshold of 0.05.

ii. On the average how much does log of productivity change with one incremental style change.

The coefficient for "no_of_style_change" is -0.0276. This means that, on average, for each incremental style change, the log of productivity decreases by approximately 0.0276 units.

iii. What is the change in log of productivity for quarter 2, 3, 4 and 5 with respect to quarter 1. Which of these changes are statistically significant?

From the provided regression output: * Quarter 2: Coefficient = -0.0028 * Quarter 3: Coefficient = -0.0365 * Quarter 4: Coefficient = -0.0531 * Quarter 5: Coefficient = 0.0817

These coefficients represent the change in log of productivity compared to the reference quarter, which is Quarter 1.

The statistical significance of these changes by examining the associated p-values: * Quarter 2: p-value = 0.882, not statistically significant. * Quarter 3: p-value = 0.086, not statistically significant. * Quarter 4: p-value = 0.010, statistically significant. * Quarter 5: p-value = 0.043, statistically significant.

Therefore, the change in log of productivity for Quarter 4 and 5 with respect to Quarter 1 is statistically significant. The changes for Quarters 2 and 3 are not statistically significant.

iv. How does the productivity of sewing department compare with the finishing department?

department_sewing: Coefficient = -0.4663

department_finishing: Coefficient = 0.1085

The coefficient for the sewing department, -0.4663, suggests a decrease in productivity compared to the reference category, indicating that employees in the sewing department tend to have lower productivity levels. Conversely, the coefficient for the finishing department, 0.1085, suggests an increase in productivity compared to the reference category, implying that employees in the finishing department tend to have higher productivity levels.

v. Write down the regression equation for the following cases:

1. Sewing department for a Sunday of quarter 4 for team 10.

log_actual_productivity= -0.6312 - 0.4663(sewing) + 0.0531(quarter_Quarter4) + 0.0171(day_Sunday) - 0.1235(team_10)

2. Finishing department for a Wednesday of quarter 1 for team 4.

log_actual_productivity= -0.6312 + 0.1085(finishing) + 0(quarter_Quarter1) + 0.0244(day_Wednesday) - 0.0405(team_4)

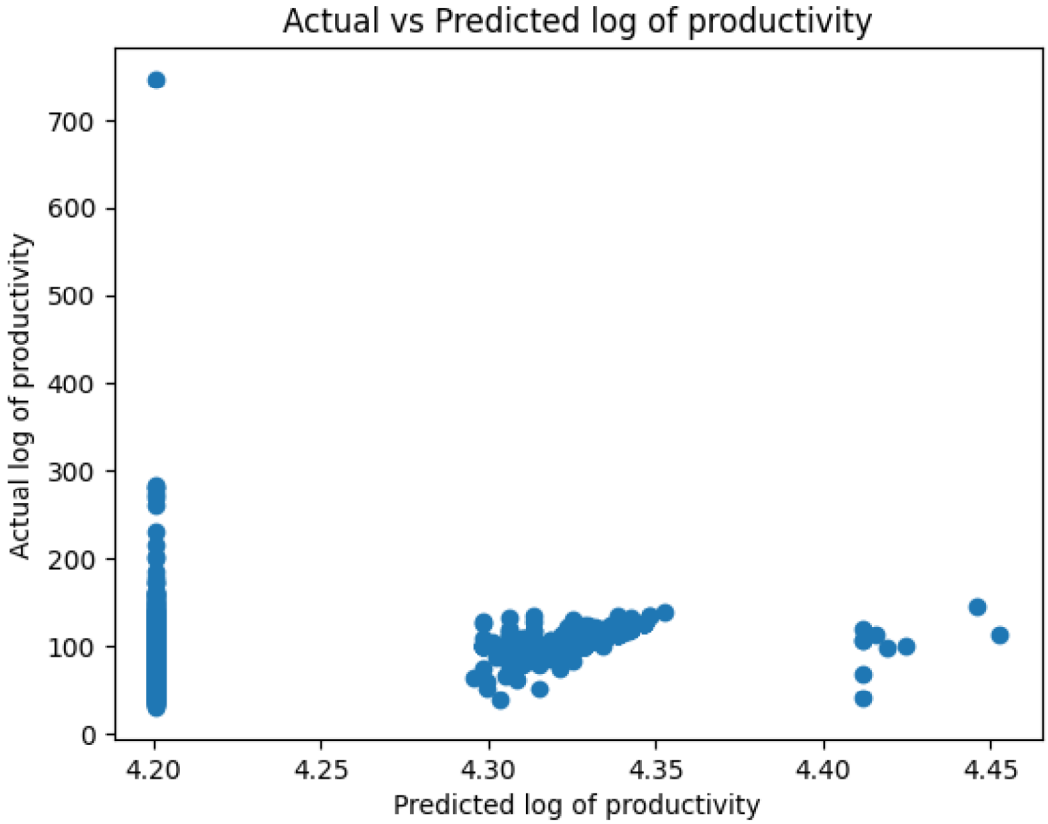
3. Finishing department for a Monday of quarter 2 for team 8.

log_actual_productivity= -0.6312 + 0.1085(finishing) - 0.0028(quarter_Quarter2) + 0(day_Monday) - 0.1212(team_8)

vi. Plot the actual log of productivity values versus the predicted log of productivity values. Do you think the model is a good fit? How much variance of the response is explained by the model?

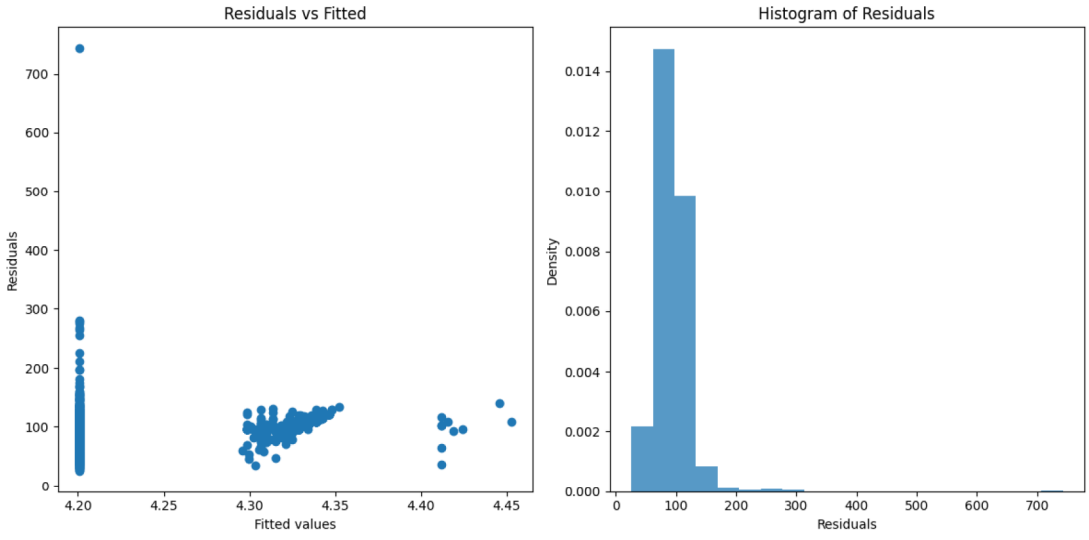
With an R-squared value of 0.325, it indicates that approximately 32.5% of the variance in the response variable (actual log productivity) is explained by the model.

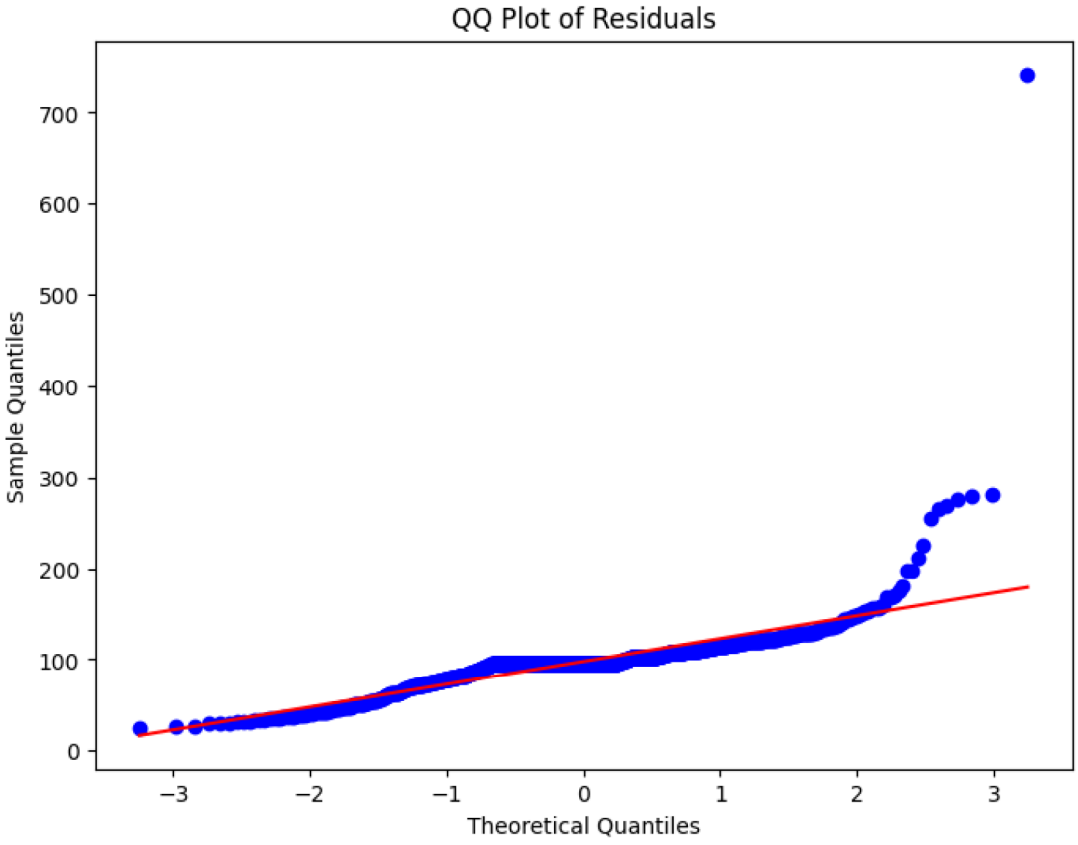
[ ]:
```
plt.scatter(predictions, y)
plt.xlabel('Predicted log of productivity')
plt.ylabel('Actual log of productivity')
plt.title('Actual vs Predicted log of productivity')
plt.show()
```



vii. Plot the residuals and the distribution of the residuals. Plot the qqnorm and qqline of the residuals.

[ ]:
```
import scipy.stats as stats

residuals = y - predictions

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.scatter(predictions, residuals)
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.title('Residuals vs Fitted')

plt.subplot(1, 2, 2)
plt.hist(residuals, bins=20, density=True, alpha=0.75)
plt.xlabel('Residuals')
plt.ylabel('Density')
plt.title('Histogram of Residuals')

plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 6))
stats.probplot(residuals, dist="norm", plot=plt)
plt.title('QQ Plot of Residuals')
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Sample Quantiles')
plt.show()
```





d. Repeat the above (c) for percentage_achievement as the response and all other predictors as above except targeted_productivity.

[ ]:
```
d['const'] = 1
X = d[['const', 'log_no_of_workers_1', 'log_incentive_1', 'no_of_style_change',␣
 ↪'quarter', 'department', 'day', 'team']]
y = d['percentage_achievement']
X = pd.get_dummies(X, columns=['quarter', 'department', 'day', 'team'],␣
 ↪drop_first=True)
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                                OLS Regression Results
==============================================================================
==
Dep. Variable:     percentage_achievement   R-squared:
0.089
Model:                               OLS   Adj. R-squared:
0.069
Method:                    Least Squares   F-statistic:
```

```
4.548
Date:                    Fri, 01 Mar 2024   Prob (F-statistic):
1.47e-12
Time:                           09:50:41   Log-Likelihood:
-5779.6
No. Observations:                1197   AIC:
1.161e+04
Df Residuals:                    1171   BIC:
1.174e+04
Df Model:                          25
Covariance Type:            nonrobust
================================================================================
=========
                             coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---------
const                     55.1257     10.629      5.187      0.000      34.273
75.979
log_no_of_workers_1       20.5411      3.987      5.152      0.000      12.718
28.364
log_incentive_1            3.5297      0.735      4.799      0.000       2.087
4.973
no_of_style_change        -2.8162      2.375     -1.186      0.236      -7.477
1.844
quarter_Quarter2           1.2177      2.356      0.517      0.605      -3.405
5.841
quarter_Quarter3          -2.7265      2.701     -1.009      0.313      -8.026
2.573
quarter_Quarter4          -2.5648      2.629     -0.976      0.329      -7.723
2.593
quarter_Quarter5          10.6438      5.122      2.078      0.038       0.595
20.692
department_finishing       7.7789      2.789      2.789      0.005       2.307
13.251
department_sweing        -42.9533      7.108     -6.043      0.000     -56.900
-29.007
day_Saturday               1.1126      3.198      0.348      0.728      -5.162
7.387
day_Sunday                -0.8360      3.072     -0.272      0.786      -6.864
5.192
day_Thursday               3.4112      3.131      1.090      0.276      -2.731
9.554
day_Tuesday                2.4796      3.066      0.809      0.419      -3.536
8.495
day_Wednesday              1.5351      3.047      0.504      0.614      -4.442
7.512
team_2                    -4.8952      4.205     -1.164      0.245     -13.145
```

```
3.354
team_3                  -1.3014      4.368     -0.298      0.766      -9.872
7.270
team_4                  -0.3038      4.271     -0.071      0.943      -8.684
8.076
team_5                  -0.8470      4.411     -0.192      0.848      -9.502
7.808
team_6                  -4.8397      4.567     -1.060      0.289     -13.800
4.121
team_7                  -5.5971      4.383     -1.277      0.202     -14.197
3.003
team_8                  -8.0005      4.236     -1.889      0.059     -16.312
0.311
team_9                 -11.0590      4.242     -2.607      0.009     -19.383
-2.735
team_10                -11.1077      4.290     -2.589      0.010     -19.525
-2.690
team_11                 -9.7205      4.456     -2.182      0.029     -18.462
-0.979
team_12                 -2.1406      4.474     -0.478      0.632     -10.919
6.638
==============================================================================
Omnibus:                     1870.749   Durbin-Watson:                   1.773
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1707293.427
Skew:                           9.115   Prob(JB):                         0.00
Kurtosis:                     187.117   Cond. No.                         67.0
==============================================================================

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

i. Which of the following variables significantly affect workers' percentage of achievement and which direction? State the level of significance.

Significantly Affecting Variables: * Log Number of Workers (log_no_of_workers_1): Positive coefficient (20.5411) with p-value < 0.001, indicating that more workers lead to higher percentage of achievement. * Log Incentive (log_incentive_1): Positive coefficient (3.5297) with p-value < 0.001, suggesting that higher incentives result in higher percentage of achievement.

- Finishing Department: Positive coefficient (7.7789) with p-value $= 0.005$, indicating higher achievement compared to reference.
- Sewing Department: Negative coefficient (-42.9533) with p-value < 0.001, indicating lower achievement compared to reference.

Non-Significant Variables: * Number of Style Changes (no_of_style_change): Insignificant coefficient (p-value $= 0.236$). * Quarter and Day Variables: No statistically significant effects observed (all p-values > 0.05).

ii. On the average how much does percentage of achievement change with one incre-

mental style change.

The coefficient for no_of_style_change is -2.8162. This means that, on average, for each additional style change, the percentage of achievement decreases by approximately 2.8162 units.

iii. What is the change in percentage of achievement for quarter 2, 3, 4 and 5 with respect to quarter 1. Which of these changes are statistically significant?

From the provided regression output: * Quarter 2: Coefficient = 1.2177 * Quarter 3: Coefficient = -2.7265 * Quarter 4: Coefficient = -2.5648 * Quarter 5: Coefficient = 10.6438

These coefficients represent the change in percentage of achievement compared to the reference quarter, which is Quarter 1.

The statistical significance of these changes by examining the associated p-values: * Quarter 2: p-value = 0.605, not statistically significant. * Quarter 3: p-value = 0.313, not statistically significant. * Quarter 4: p-value = 0.329, not statistically significant. * Quarter 5: p-value = 0.038, statistically significant.

Therefore, the change in percentage of achievement for Quarter 5 with respect to Quarter 1 is statistically significant. The changes for Quarters 2, 3 and 4 are not statistically significant.

iv. How does the percentage of achievement of sewing department compare with the finishing department?

department_sewing: Coefficient = -42.9533

department_finishing: Coefficient = 7.7789

- Sewing Department: The coefficient for the sewing department is -42.9533, indicating a negative effect on the percentage of achievement. This means that, on average, the percentage of achievement is lower for employees in the sewing department compared to the reference category.
- Finishing Department: The coefficient for the finishing department is 7.7789, indicating a positive effect on the percentage of achievement. This suggests that, on average, the percentage of achievement is higher for employees in the finishing department compared to the reference category.

v. Write down the regression equation for the following cases:

1. Sewing department for a Sunday of quarter 4 for team 10.

percentage of achievement= 55.1257 - 42.9533(sewing) -2.5648(quarter_Quarter4) - 0.8360(day_Sunday) - 11.1077(team_10)

2. Finishing department for a Wednesday of quarter 1 for team 4.

percentage of achievement= 55.1257 + 7.7789(finishing) + 0(quarter_Quarter1) + 1.5351(day_Wednesday) - 0.3038(team_4)

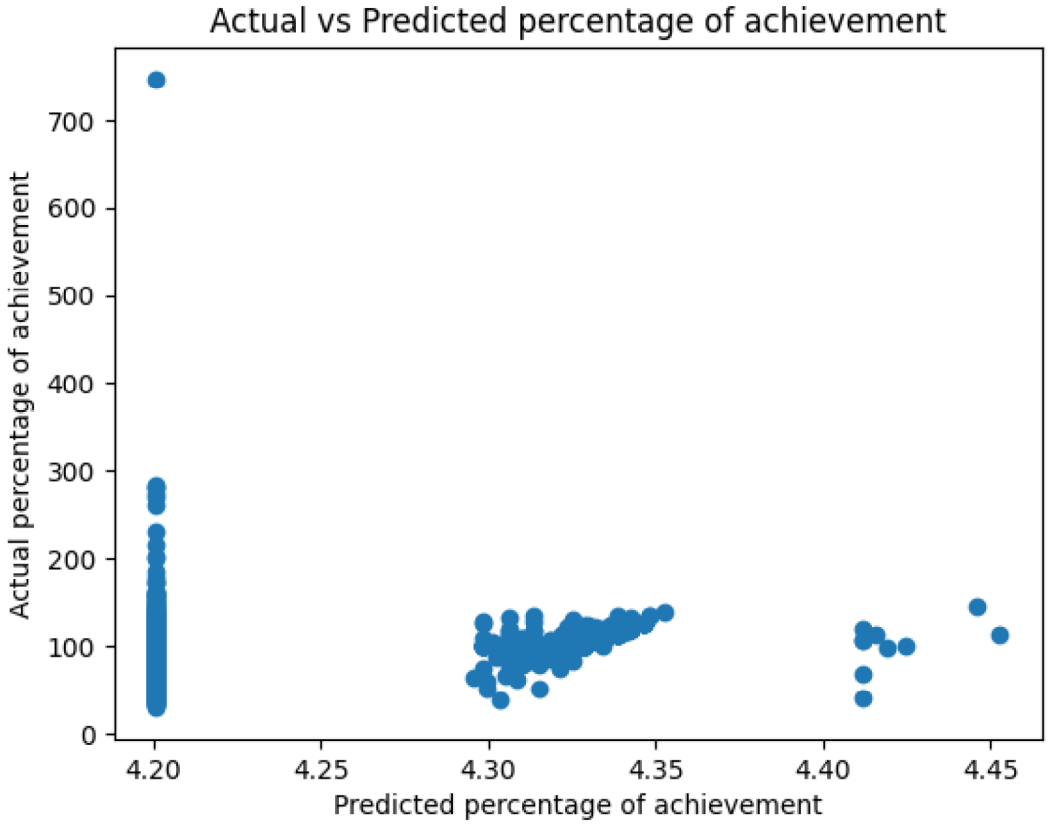
3. Finishing department for a Monday of quarter 2 for team 8.

percentage of achievement= 55.1257 + 7.7789(finishing) +1.2177(quarter_Quarter2) + 0(day_Monday) - 8.0005(team_8)

vi. Plot the actual percentage of achievement values versus the predicted percentage of achievement values. Do you think the model is a good fit? How much variance of the response is explained by the model?

With an R-squared value of 0.089, it indicates that approximately 8.9% of the variance in the response variable (percentage of achievement) is explained by the model.

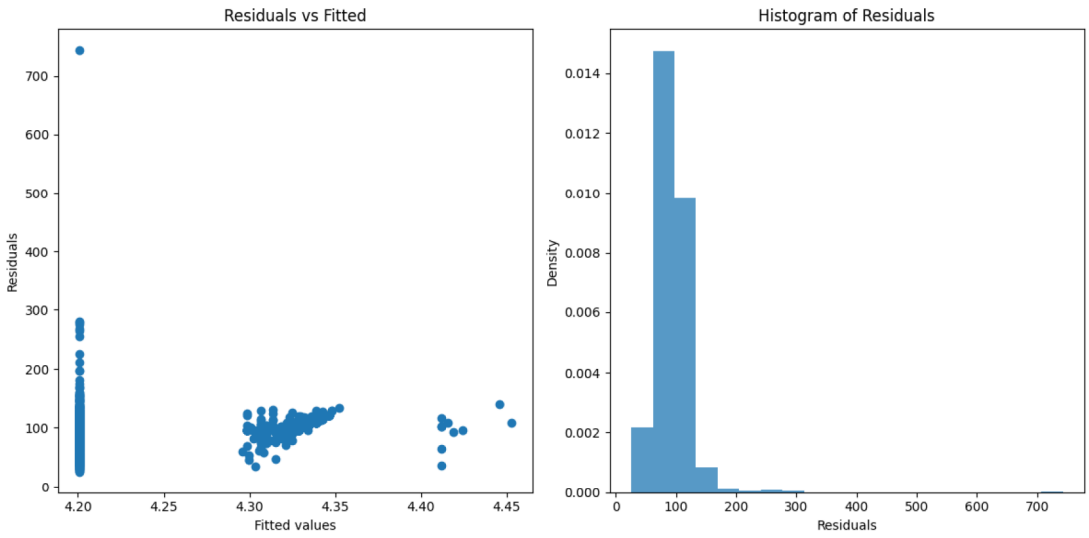
[ ]:
```
plt.scatter(predictions, y_percentage)
plt.xlabel('Predicted percentage of achievement')
plt.ylabel('Actual percentage of achievement')
plt.title('Actual vs Predicted percentage of achievement')
plt.show()
```

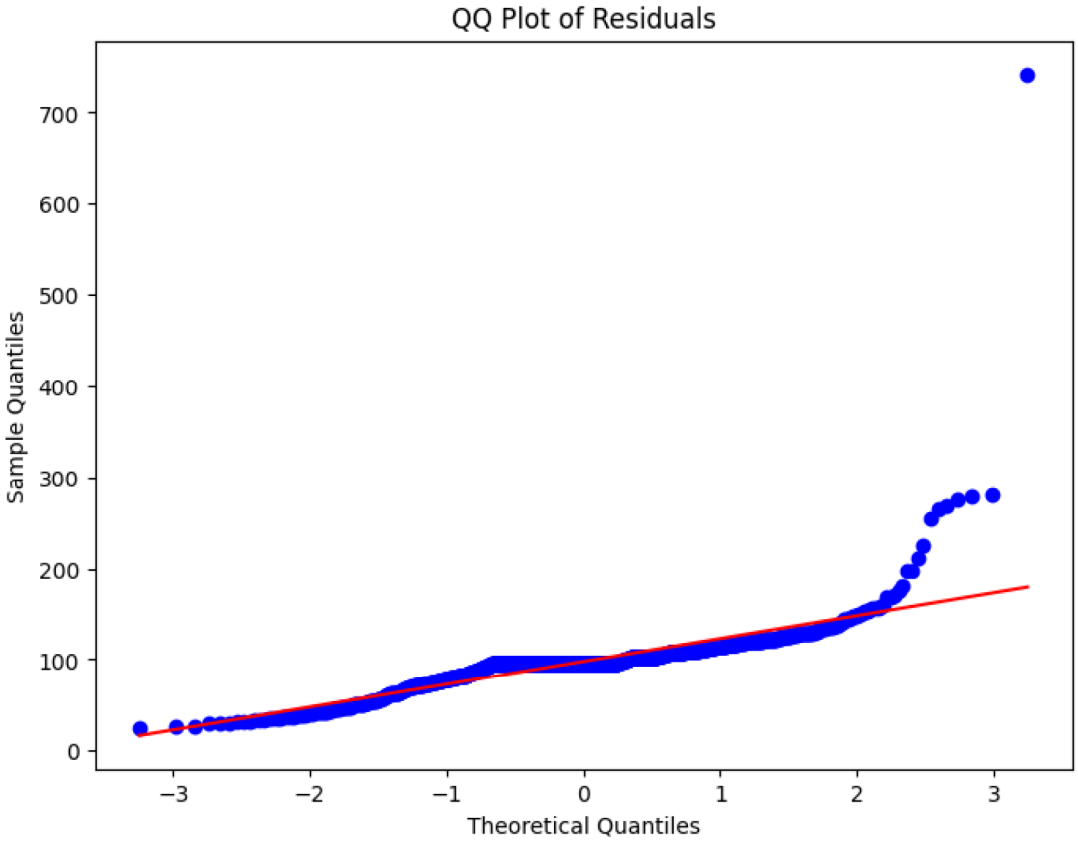


vii. Plot the residuals and the distribution of the residuals. Plot the qqnorm and qqline of the residuals.

[ ]:
```
residuals_persentage = y_percentage - predictions

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.scatter(predictions, residuals_persentage)
```

```
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.title('Residuals vs Fitted')

plt.subplot(1, 2, 2)
plt.hist(residuals_persentage, bins=20, density=True, alpha=0.75)
plt.xlabel('Residuals')
plt.ylabel('Density')
plt.title('Histogram of Residuals')

plt.tight_layout()
plt.show()

plt.figure(figsize=(8, 6))
stats.probplot(residuals_persentage, dist="norm", plot=plt)
plt.title('QQ Plot of Residuals')
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Sample Quantiles')
plt.show()
```





e. Conduct an ANOVA analysis for question (c) and explain how much (and statistical significance) variance is explained by each variables? Which variable explains the maximum variance?

Quarter, Team, Log_no_of_workers_1, Log_incentive_1, and Log_targeted_productivity have low p-values ($p < 0.05$), indicating statistical significance. Among these, Log_incentive_1 stands out with the highest F-statistic and lowest p-value, suggesting it explains the most variance. However, No_of_style_change and Department do not significantly predict log_actual_productivity. Overall, Log_incentive_1 emerges as the primary driver of productivity variance in the model.

[ ]:
```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
model = ols('log_actual_productivity ~ log_no_of_workers_1 + log_incentive_1 +␣
 ↪log_targeted_productivity + no_of_style_change + quarter + department + day␣
 ↪+ team', data=d).fit()
anova_results = anova_lm(model)
print(anova_results)
```

```
                             df      sum_sq    mean_sq           F  \
quarter                     4.0    1.896123   0.474031    8.200142
department                  2.0    1.058344   0.529172    9.154013
```

day	5.0	0.182351	0.036470	0.630887
team	11.0	7.189665	0.653606	11.306566
log_no_of_workers_1	1.0	1.940546	1.940546	33.569021
log_incentive_1	1.0	12.707431	12.707431	219.822698
log_targeted_productivity	1.0	7.421932	7.421932	128.390152
no_of_style_change	1.0	0.125637	0.125637	2.173371
Residual	1170.0	67.634939	0.057808	NaN

	PR(>F)
quarter	1.602162e-06
department	1.135655e-04
day	6.762223e-01
team	3.393111e-20
log_no_of_workers_1	8.833658e-09
log_incentive_1	1.056155e-45
log_targeted_productivity	2.591534e-28
no_of_style_change	1.406872e-01
Residual	NaN

4. Managerial Insights – 10 Points.

Summarize your findings from the above analysis. What can managers of garment manufacturing units learn from your analysis of the data? If a manager is interested in improving the productivity of a garment manufacturing unit, what actions would you suggest (reasonable actions, you cannot ask to stop functioning of a division) to adopt?

Based on the analysis provided, managers can gain insights into the distribution of data through boxplots and scatterplot regression. Additionally, they can utilize t-tests to assess the significance of variables affecting worker productivity and their directional impact.

Based on the analysis of the data, managers of garment manufacturing units can derive several key insights to improve productivity:

- Identify High-Performing Periods: Quarter 5 exhibits higher productivity compared to other quarters, as evidenced by a statistically significant p-value below 0.05. Managers should investigate the factors contributing to this high productivity period and consider implementing similar strategies in other quarters.
- Consider Style Changes Carefully: The analysis suggests that productivity tends to be lower when there are style changes. Managers should carefully assess the impact of style changes on productivity and consider strategies to minimize disruptions during these periods.
- Address Departmental Inefficiencies: Among all departments, the Sewing department shows the lowest productivity. To address this issue, managers can focus on strengthening staff training within the Sewing department and identify and address any inefficiencies contributing to lower productivity.
- Evaluate the Impact of Incentives: Although the correlation between incentives (log_incentive_1) and productivity is not very strong, it remains statistically significant. While incentives may not explain a significant portion of productivity variance, managers

should still consider the potential impact of incentives on productivity and explore ways to optimize incentive programs.

Overall, managers should use a data-driven approach to identify areas for improvement and implement targeted strategies to enhance productivity. This may include optimizing production schedules, providing targeted training and support to departments with lower productivity, and continuously evaluating and refining incentive programs to maximize their effectiveness.