```
[1] "DATA CLEANED AND DATAFRAME FINAL MERGED"
[1] "DATA CLEANED"
[1] "DATA CLEANED AND DATAFRAME"
[1] "DATA CLEANED AND DATAFRAME"
[1] "DATA CLEANED AND DATAFRAME"
[1] "DATA CLEANED AND DATAFRAME"
Rows(sample_size) variable_name(missing_count)
Rows(sample_size) variable_name(missing_count)
Rows(sample_size) variable_name(missing_count)
```

3.7 - WIP

```
Work in progress. Its total efficiency sucks.
Idea: Calculate the ratio between sms received and sms sent.
Then calculate the ratio between productivity and the ratio.
Then find the ratio of the ratio of productivity and the ratio.
```

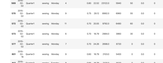
```
df_wip.head(20)
```

	year	quarter	department	day_name	team	targeted productivity	sms	wip	over_time	incentive	idle_time	idle_men	no_of_style_change
0	2015	Quarter1	sewing	Thursday	8	0.80	26.16	0.032692307692307692	7080	113	0.0	0.0	0
1	2015	Quarter1	sewing	Friday	1	0.80	23.00	0.028749999999999999	960	98	0.0	0.0	0
2	2015	Quarter1	sewing	Saturday	11	0.80	23.419999999999999	0.029274999999999999	3660	50	0.0	0.0	0
3	2015	Quarter1	sewing	Monday	12	0.80	15.26	0.019075000000000000	3062	50	0.0	0.0	0
4	2015	Quarter1	sewing	Tuesday	10	0.80	15.26	0.019075000000000000	3062	50	0.0	0.0	0
5	2015	Quarter1	sewing	Wednesday	2	0.80	10.00	0.012500000000000000	1800	30	0.0	0.0	0
6	2015	Quarter1	sewing	Thursday	8	0.75	19.12	0.025466666666666667	3060	45	0.0	0.0	0
7	2015	Quarter1	sewing	Friday	1	0.75	19.12	0.025466666666666667	3060	45	0.0	0.0	0
8	2015	Quarter1	sewing	Saturday	11	0.75	29.94	0.039920000000000000	4800	70	0.0	0.0	0
9	2015	Quarter1	sewing	Monday	5	0.70	16.79	0.023985714285714286	3360	50	0.0	0.0	0
10	2015	Quarter1	sewing	Tuesday	3	0.70	16.79	0.023985714285714286	3360	50	0.0	0.0	0

3.8 - Over Time

```
Time is and of time the faster over time is faster.
```

```
df_over_time.head(20)
```



3.9 - Productivity

Actual productivity in days of the week.

```
Saturday has the highest productivity.
```

```
df_actual_productivity_by_day_of_week=df[['actual_productivity','day_name']].groupby('day_name').mean().sort_values('actual_productivity',ascending=False).reset_index()
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_actual_productivity_by_day_of_week,
x='day_name',
y='actual_productivity',
ax=ax,
hue='day_name',
legend=False,
palette='Pastel1')
ax.set_xlabel('Day of the week')
ax.set_ylabel('Actual productivity')
ax.set_title('Actual productivity by day of the week')
plt.show()
```


Actual productivity in months.

```
Saturday has the highest productivity month.
```

```
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_actual_productivity_by_month,
x='month',
y='actual_productivity',
ax=ax,
hue='month',
legend=False,
palette='Pastel1')
ax.set_xlabel('Month')
ax.set_ylabel('Actual productivity')
ax.set_title('Actual productivity by month')
plt.show()
```

Actual productivity in quarters.

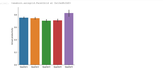
```
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_actual_productivity_by_quarter,
x='quarter',
y='actual_productivity',
ax=ax,
hue='quarter',
legend=False,
palette='Pastel1')
ax.set_xlabel('Quarter')
ax.set_ylabel('Actual productivity')
ax.set_title('Actual productivity by quarter')
plt.show()
```

Actual productivity by teams

```
df_actual_productivity_by_team=df[['actual_productivity','team']].groupby('team').mean().sort_values('actual_productivity',ascending=False).reset_index()
df_count=df[['actual_productivity','team']].groupby('team').count().reset_index()
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_actual_productivity_by_team,
x='team',
y='actual_productivity',
ax=ax,
hue='actual_productivity',
legend=False,
palette='Pastel1')
for p,q in zip(ax.patches,df_count['team']):
ax.text(p.get_x()+p.get_width()/2,
p.get_height()+0.0001,
q,
ha='center',
va='bottom',
fontsize=6,
rotation=90)
ax.set_xlabel('Team number')
ax.set_ylabel('Actual productivity')
ax.set_title('Actual productivity by teams')
plt.show()
```

Target productivity by teams

```
df_target_productivity_by_team=df[['targeted_productivity','team']].groupby('team').mean().sort_values('targeted_productivity',ascending=False).reset_index()
df_count=df[['targeted_productivity','team']].groupby('team').count().reset_index()
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_target_productivity_by_team,
x='team',
y='targeted_productivity',
ax=ax,
hue='targeted_productivity',
legend=False,
palette='Pastel1')
for p,q in zip(ax.patches,df_count['team']):
ax.text(p.get_x()+p.get_width()/2,
p.get_height()+0.0001,
q,
ha='center',
va='bottom',
fontsize=6,
rotation=90)
ax.set_xlabel('Team number')
ax.set_ylabel('Targeted productivity')
ax.set_title('Target productivity by teams')
plt.show()
```



Distribution of Targeted productivity

```
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.histplot(df_targeted_productivity, data=df, color_palette=sns.color_palette("Pastel1",n_colors=1),
kde=False,
stat="count",
bins=[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
ax=ax,
hue=None,
multiple="stack",
element="bars",
fill=True,
shrink=0.9,
discrete=False,
cumulative=False,
log_scale=False,
stacked=False,
kde_kws={'bw_adjust': 1},
line_kws={'linewidth': 1.5,
'linestyle': '-',
'antialiased': True},
legend=False)
ax.set_xlabel('Targeted Productivity')
ax.set_ylabel('Count')
ax.set_title('Distribution of Targeted productivity')
plt.show()
```

Distribution of Actual productivity

```
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.histplot(df_actual_productivity, data=df, color_palette=sns.color_palette("Pastel1",n_colors=1),
kde=False,
stat="count",
bins=[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
ax=ax,
hue=None,
multiple="stack",
element="bars",
fill=True,
shrink=0.9,
discrete=False,
cumulative=False,
log_scale=False,
stacked=False,
kde_kws={'bw_adjust': 1},
line_kws={'linewidth': 1.5,
'linestyle': '-',
'antialiased': True},
legend=False)
ax.set_xlabel('Actual Productivity')
ax.set_ylabel('Count')
ax.set_title('Distribution of Actual productivity')
plt.show()
```

Actual productivity vs Targeted productivity

```
plt.figure(figsize=(7.5,3.5))
ax=sns.lineplot(df_productivity_by_d,
x='day',
y='productivity',
hue='productivity',
palette='Pastel1',
linewidth=1.5,
legend=False)
ax.set_xlabel('Day')
ax.set_ylabel('Productivity')
ax.set_title('Actual productivity vs Targeted productivity')
plt.show()
```

3.10 - Incentive

Overview

```
df_incentive_over_time=df[['actual_productivity','no_of_workers','incentive','over_time','team','department']].sample(1000).reset_index()
sns.pairplot(df_incentive_over_time,
hue='team',
palette='Pastel1',
diag_kind='hist',
corner=True,
vars=['actual_productivity','no_of_workers','incentive','over_time'])
plt.show()
```

```
df_incentive_over_time.corr()
```

```
                       actual_productivity  no_of_workers  incentive  over_time
actual_productivity               1.000000       0.000000   0.000000   0.000000
no_of_workers                     0.000000       1.000000   0.000000   0.000000
incentive                         0.000000       0.000000   1.000000   0.000000
over_time                         0.000000       0.000000   0.000000   1.000000
```

Incentive heat map with other attributes

```
df_incentive_heat_map=df[['actual_productivity','no_of_workers','incentive','over_time','team','department']].drop_duplicates()
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
sns.heatmap(df_incentive_heat_map, annot=True,
cmap='Pastel1',
ax=ax,
linewidth=0.5,
cbar=True,
square=False,
fmt=".2f",
annot_kws={'size': 6,
'weight': 'normal',
'rotation': 0},
xticklabels=True,
yticklabels=True,
cbar_kws={'shrink': 0.9,
'orientation': 'vertical',
'pad': 0.05,
'aspect': 20,
'ticks': [0.8, 0.6, 0.4, 0.2, 0.0],
'extend': 'neither',
'extendfrac': 0.05,
'drawedges': False})
ax.set_xlabel('Attributes')
ax.set_ylabel('Attributes')
ax.set_title('Incentive heat map with other attributes')
plt.show()
```

```
                       actual_productivity  no_of_workers  incentive  over_time
actual_productivity               1.000000       0.000000   0.000000   0.000000
no_of_workers                     0.000000       1.000000   0.000000   0.000000
incentive                         0.000000       0.000000   1.000000   0.000000
over_time                         0.000000       0.000000   0.000000   1.000000
```

```
df_incentive_over_time.corr(numeric_only=True).T
```

```
                       actual_productivity  no_of_workers  incentive  over_time
actual_productivity               1.000000       0.000000   0.000000   0.000000
no_of_workers                     0.000000       1.000000   0.000000   0.000000
incentive                         0.000000       0.000000   1.000000   0.000000
over_time                         0.000000       0.000000   0.000000   1.000000
```

Incentive paid to each team with respect to actual productivity

```
import pandas as pd
df_incentive_actual_productivity_by_team=df[['actual_productivity','incentive','team']].groupby('team').mean().sort_values('actual_productivity',ascending=False).reset_index()
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_incentive_actual_productivity_by_team,
x='team',
y='incentive',
ax=ax,
hue='incentive',
legend=False,
palette='Pastel1')
ax.set_xlabel('Team number')
ax.set_ylabel('Incentive')
ax.set_title('Incentive paid to each team')
plt.show()
```

Incentive paid to each team with respect to actual productivity

```
import pandas as pd
df_incentive_actual_productivity_by_team=df[['actual_productivity','incentive','team']].groupby('team').mean().sort_values('actual_productivity',ascending=False).reset_index()
fig,ax=plt.subplots(1,1,figure=True,dpi=300)
ax=sns.barplot(df_incentive_actual_productivity_by_team,
x='actual_productivity',
y='incentive',
ax=ax,
hue='incentive',
legend=False,
palette='Pastel1')
ax.set_xlabel('Actual productivity')
ax.set_ylabel('Incentive')
ax.set_title('Incentive paid to each team w.r.t actual productivity')
plt.show()
```

Incentive paid to workers

```
# plot:incentive_paid_to_workers_actual_productivity_by_month
```

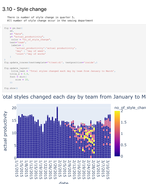
```
# log_df_dbl
df = log_df_dbl
df1 = df[df$department == "finishing"]
df2 = df[df$department == "sewing"]
df1 = df1[,c("actual_productivity")]
df2 = df2[,c("actual_productivity")]
df = cbind(df1,df2)
colnames(df) = c("finishing","sewing")
df = melt(df, measure.vars = c("finishing","sewing"))
ggplot(df,aligned=TRUE,
       aes(x = value, fill = variable)) +
  geom_density(
    alpha = 0.5,
    bins = 30,
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density",
    na.rm = FALSE,
    orientation = "x",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    stat = "density",
    position = "identity",
    na.rm = FALSE,
    orientation = "x",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_by_department.jpeg",dpi=300)
```

Actual productivity of sewing and finishing department

```
# plot:actual_productivity_by_day_of_the_week
log_df_dbl = log_df_dbl[log_df_dbl$quarter != "Quarter5"]
ggplot(log_df_dbl,aligned=TRUE,
       aes(x = actual_productivity,
           y = day_of_week,
           fill = stat(quantile),
           group = day_of_week)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_by_day_of_the_week.jpeg",dpi=300)
```

```
# log_df_dbl
log_df_dbl = log_df_dbl[log_df_dbl$quarter != "Quarter5"]
ggplot(log_df_dbl,aligned=TRUE,
       aes(x = actual_productivity,
           y = month,
           fill = stat(quantile),
           group = month)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_by_month.jpeg",dpi=300)
```

```
# log_df_dbl
log_df_dbl = log_df_dbl[log_df_dbl$quarter != "Quarter5"]
ggplot(log_df_dbl,aligned=TRUE,
       aes(x = actual_productivity,
           y = team,
           fill = stat(quantile),
           group = team)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_by_team.jpeg",dpi=300)
```



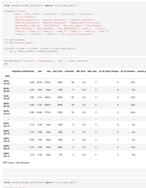

```
# log_df_dbl
log_df_dbl = log_df_dbl[log_df_dbl$quarter != "Quarter5"]
ggplot(log_df_dbl,aligned=TRUE,
       aes(x = actual_productivity,
           y = team,
           fill = stat(quantile),
           group = team)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_of_team_with_different_diversity_in_product_and_members.jpeg",dpi=300)
```

3.10 - Style change

```
# Actual productivity of style change by quarter 1,
# Actual productivity of style change in the sewing department
```

```
# log_df_dbl
df = log_df_dbl
df = df[,c("actual_productivity",
           "no_of_style_change",
           "quarter",
           "department")]
df = df[df$department == "sewing"]
df = df[df$quarter == "Quarter1"]
df = df[,c("actual_productivity",
           "no_of_style_change")]
ggplot(df,aligned=TRUE,
       aes(x = actual_productivity,
           y = no_of_style_change,
           fill = stat(x),
           group = no_of_style_change)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_of_sewing_in_quarter1_based_on_no_of_style_change.jpeg",dpi=300)
```

```
# log_df_dbl
df = log_df_dbl
df = df[,c("actual_productivity",
           "no_of_style_change",
           "department")]
df = df[df$department == "sewing"]
df = df[,c("actual_productivity",
           "no_of_style_change")]
ggplot(df,aligned=TRUE,
       aes(x = actual_productivity,
           y = no_of_style_change,
           fill = stat(x),
           group = no_of_style_change)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_of_sewing_in_quarter1_based_on_no_of_style_change.jpeg",dpi=300)
```




```
# log_df_dbl
ggplot(log_df_dbl,aligned=TRUE,
       aes(x = actual_productivity,
           y = no_of_style_change,
           fill = stat(x),
           group = no_of_style_change)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_of_sewing_in_quarter1_based_on_no_of_style_change.jpeg",dpi=300)
```

```
# log_df_dbl
df = log_df_dbl
df = df[,c("actual_productivity",
           "no_of_style_change",
           "department")]
df = df[df$department == "finishing"]
df = df[,c("actual_productivity",
           "no_of_style_change")]
ggplot(df,aligned=TRUE,
       aes(x = actual_productivity,
           y = no_of_style_change,
           fill = stat(x),
           group = no_of_style_change)) +
  stat_density_ridges(
    quantile_lines = TRUE,
    calc_ecdf = TRUE,
    geom = "density_ridges_gradient",
    position = "identity",
    show.legend = TRUE,
    inherit.aes = TRUE,
    stat = "density_ridges",
    na.rm = FALSE,
    orientation = "y",
    bw = "nrd0",
    adjust = 1,
    kernel = "gaussian",
    trim = FALSE,
    bounds = c(-Inf, Inf),
    quantile = 2,
    outline.type = "upper",
    lineend = "butt",
    linejoin = "mitre",
    linewidth = NULL,
    linetype = 1,
    color = NA,
    fill = "grey20",
    weight = NULL,
    show.legend = NA,
    inherit.aligned = TRUE,
    ...
    )
# ggsave("actual_productivity_of_finishing_in_quarter1_based_on_no_of_style_change.jpeg",dpi=300)
```

4 - One-hot encoding

- we use following scheme to product productivity range:
  - Low
  - Medium
  - High
  - Very_high
  - Extreme

```
# 4 - One-hot encoding
# we use following scheme to product productivity range:
# Low
# Medium
# High
# Very_high
# Extreme
log_df = fread("processed_productivity.csv")
log_df = log_df[,c("actual_productivity","quarter","department","day","no_of_workers","no_of_style_change")]
```