walmart

September 18, 2022

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
```

[3]:
```
from google.colab import drive

drive.mount("/content/drive")
```

```
Mounted at /content/drive
```

[4]:
```
data=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/walmart_data.txt")
```

[5]:
```
data.head()
```

[5]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969

[6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   User_ID                     550068 non-null  int64
```

```
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64
 8   Product_Category            550068 non-null  int64
 9   Purchase                    550068 non-null  int64
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

1 Conclusion

1.0.1 no null value

2 Need to change data formats required for convinence

[7]:
```
data["City_Category"]=data.City_Category.replace({"A":1,"B":2,"C":3})
# converting city cateogry to 1,2,3 for int format
```

[8]:
```
#converting user id to obj
data["User_ID"]=data.User_ID.astype("object")
```

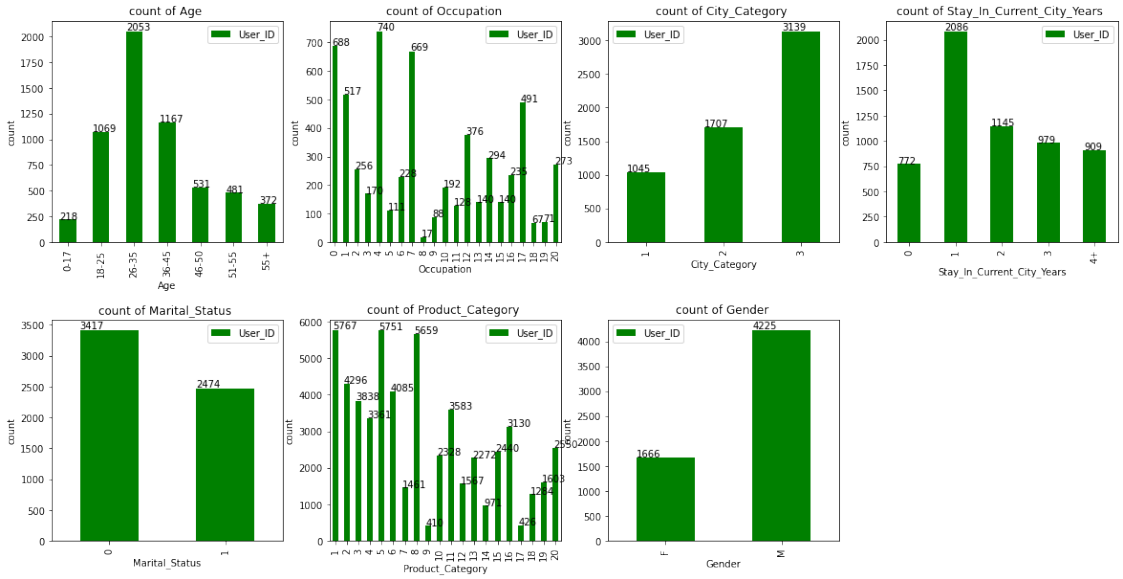
[9]:
```
data.head()
```

[9]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	1	2	0	3	8370
1	1000001	P00248942	F	0-17	10	1	2	0	1	15200
2	1000001	P00087842	F	0-17	10	1	2	0	12	1422
3	1000001	P00085442	F	0-17	10	1	2	0	12	1057
4	1000002	P00285442	M	55+	16	3	4+	0	8	7969

[10]:
```
column=["Age","Occupation","City_Category","Stay_In_Current_City_Years","Marital_Status","Prod
count=0
for i in column:
  count+=1
  plt.subplot(2,4,(count))
```

```
 column_ratio=data.groupby(i)["User_ID"].nunique()
 plt.subplots_adjust(hspace=0.35)
 ax=column_ratio.plot(kind="bar",ylabel="count",title=f"count of {i}␣
↪",legend=True,figsize=(20,10),color="g")
 for p in ax.patches:
     ax.annotate(str(p.get_height()), (p.get_x() * 1.005, p.get_height() * 1.
↪005))
```

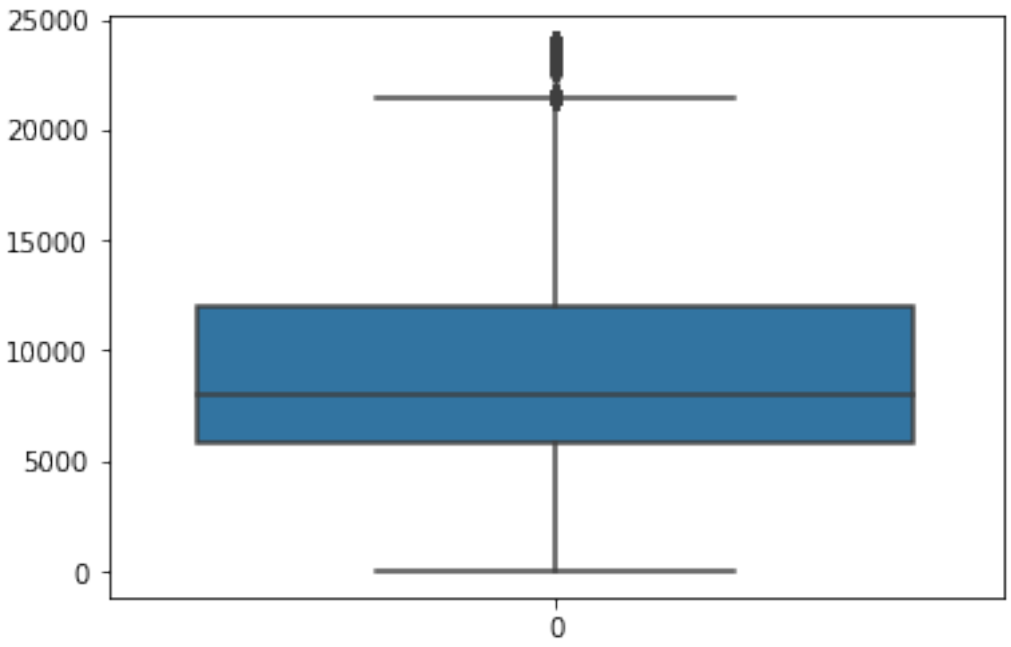


[11]:
```
data.columns
```

[11]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

[12]:
```
#Detecting outliers with box plot
sns.boxplot(data=data.Purchase)
```

[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f358b7acb10>
```



[13]:
```
data.groupby("Gender")["Purchase"].mean()
```

[13]:
```
Gender
F    8734.565765
M    9437.526040
Name: Purchase, dtype: float64
```

By looking at a sample we can see there is a difference in purchase behaviour according to gender, since it is just a sample we will proceed to apply clt to verify behaviour over population

[14]:
```
d=np.array(data[data["Gender"]=="M"].Purchase) #making numpy array of male␣
↪purchase
d
```

[14]:
```
array([ 7969, 15227, 19215, …,   494,   473,   368])
```

[15]:
```
sample_mean=np.random.choice(d,size=300,replace=True).mean() # finding sample␣
↪mean of male purchase with replacement and sample size 200
sample_mean
```

[15]:
```
9098.963333333333
```

[16]:
```
# making list of various sample_means of 200 in sample size for 1000 times
population_mean_list=[np.random.choice(d,size=300,replace=True).mean() for i in␣
↪range (1000)]
print(population_mean_list)
```

[9232.543333333333, 9595.393333333333, 9385.233333333334, 9651.4,
9494.663333333334, 9595.166666666666, 9366.81, 9516.82, 9767.296666666667,
9375.276666666667, 8998.173333333334, 9562.63, 9412.993333333334,
9159.533333333333, 8945.49, 9210.083333333334, 9195.503333333334,
9673.086666666666, 9635.043333333333, 9555.876666666667, 9332.983333333334,
9655.156666666666, 9392.283333333333, 10067.58, 9660.03, 9612.333333333334,
9614.576666666666, 9073.353333333333, 9588.0, 9881.96, 9287.466666666667,
9448.306666666667, 9494.553333333333, 10069.103333333333, 9561.443333333333,
9521.636666666667, 9804.46, 9323.363333333333, 9860.273333333333, 9711.27,
9794.47, 9666.97, 9329.47, 9392.623333333333, 9282.686666666666,
9584.886666666667, 9297.94, 9379.656666666666, 9277.876666666667,
8970.023333333333, 9727.433333333332, 9542.06, 9074.856666666667,
9430.086666666666, 9568.426666666666, 9898.37, 9487.386666666667,
9619.253333333334, 8763.37, 9232.083333333334, 9296.103333333333,
9876.966666666667, 8807.083333333334, 9293.553333333333, 9053.696666666667,
9484.12, 9465.19, 9156.85, 9295.346666666666, 9293.92, 9155.556666666667,
9452.026666666667, 9847.433333333332, 9622.026666666667, 9470.163333333334,
9410.84, 9776.046666666667, 9636.973333333333, 9277.15, 9957.253333333334,
9201.07, 9791.55, 9860.393333333333, 9271.256666666666, 9080.163333333334,
9364.473333333333, 9591.793333333333, 9818.226666666667, 9228.66,
9920.363333333333, 8910.79, 9728.22, 9814.46, 9823.423333333334,
8721.243333333334, 9328.09, 9480.726666666667, 9915.513333333334,
9727.806666666667, 9248.29, 9731.673333333334, 8946.486666666666, 9573.98,
9200.21, 9459.983333333334, 9592.78, 9926.523333333333, 9430.346666666666,
9654.92, 9503.46, 9361.816666666668, 8554.666666666666, 9296.07,
9913.683333333332, 9761.06, 8932.52, 9547.796666666667, 8944.006666666666,
9002.343333333334, 9699.446666666667, 9260.08, 9399.446666666667, 9809.55,
8722.033333333333, 9254.673333333334, 9337.146666666667, 9050.49,
9224.003333333334, 9902.14, 8932.343333333334, 9656.776666666667, 8898.33,
9799.806666666667, 9537.096666666666, 9455.24, 10126.09, 9427.446666666667,
8870.213333333333, 9660.56, 10418.0, 9331.483333333334, 9313.26,
9750.766666666666, 9733.326666666666, 9054.913333333334, 9626.21,
9659.776666666667, 9732.996666666666, 9804.606666666667, 9515.05, 9143.96,
9753.586666666666, 9298.643333333333, 9325.563333333334, 9364.516666666666,
9896.676666666666, 9698.116666666667, 9782.676666666666, 9678.883333333333,
9603.303333333333, 9867.116666666667, 9450.33, 9158.64, 10097.85,
9227.146666666667, 9258.553333333333, 9539.6, 9786.046666666667,
9493.416666666666, 9581.36, 9551.633333333333, 9121.706666666667, 9532.33,
9265.166666666666, 9432.113333333333, 9704.906666666666, 9308.766666666666,
9969.396666666667, 9309.92, 9598.806666666667, 9605.596666666666, 9981.45,
9343.666666666666, 9058.48, 9442.95, 9399.54, 9560.243333333334, 9917.56,
9418.876666666667, 9583.263333333334, 9437.346666666666, 9870.21,
9245.026666666667, 9817.5, 9847.093333333334, 9207.206666666667,
9476.383333333333, 9248.453333333333, 9113.036666666667, 8855.676666666666,
9148.663333333334, 9487.46, 9475.573333333334, 9251.23, 9196.91,
9360.016666666666, 9719.0, 9576.14, 9292.913333333334, 9269.906666666666,
9426.39, 9312.87, 9415.43, 9273.423333333334, 9225.41, 8704.443333333333,
9377.753333333334, 9787.73, 9451.926666666666, 9802.72, 9587.15,

9660.353333333333, 9448.403333333334, 9271.926666666666, 9028.37,
9038.626666666667, 9426.273333333333, 9346.166666666666, 9452.873333333333,
9778.056666666667, 9315.06, 9358.38, 9521.66, 9552.86, 9203.823333333334,
9436.15, 9656.093333333334, 9290.596666666666, 9225.626666666667,
9125.803333333333, 9648.106666666667, 9877.486666666666, 9487.646666666667,
9316.33, 9605.13, 9538.22, 9157.57, 9320.96, 9195.086666666666, 9874.85,
10178.643333333333, 9189.223333333333, 8840.846666666666, 9844.013333333334,
9754.083333333334, 9339.716666666667, 9711.653333333334, 9669.023333333333,
9468.163333333334, 9585.093333333334, 9984.3, 9655.116666666667, 9059.89,
9360.376666666667, 9305.88, 9517.236666666666, 9764.613333333333,
9826.983333333334, 9074.106666666667, 9388.533333333333, 9743.896666666667,
9483.596666666666, 9876.066666666668, 9367.486666666666, 9854.283333333333,
9361.816666666668, 8956.876666666667, 8765.753333333334, 9641.286666666667,
9740.69, 9388.06, 9345.5, 9535.39, 9650.856666666667, 9406.433333333332,
9120.97, 9339.416666666666, 9688.72, 9181.91, 9022.363333333333,
9371.186666666666, 9422.483333333334, 9741.89, 9520.186666666666,
9411.066666666668, 9371.94, 9808.473333333333, 9578.486666666666,
9524.206666666667, 9216.093333333334, 9553.806666666667, 9567.703333333333,
9107.806666666667, 9299.616666666667, 9659.293333333333, 9355.903333333334,
9019.186666666666, 9632.826666666666, 9514.06, 9184.506666666666, 9055.81,
9264.643333333333, 9372.133333333333, 9348.706666666667, 9691.523333333333,
9241.506666666666, 9713.576666666666, 9715.503333333334, 9141.446666666667,
9334.756666666666, 9279.463333333333, 9585.12, 9204.126666666667,
9185.296666666667, 9402.276666666667, 9362.476666666667, 9494.446666666667,
9437.58, 8933.106666666667, 9248.966666666667, 9627.683333333332,
8505.926666666666, 9699.986666666666, 9789.91, 9813.983333333334,
9587.553333333333, 9217.11, 9177.89, 9946.493333333334, 9148.093333333334,
9277.763333333334, 9204.546666666667, 9602.22, 8909.116666666667,
9565.506666666666, 9277.98, 9244.813333333334, 9670.886666666667, 9277.75,
9334.786666666667, 9449.116666666667, 9473.286666666667, 9521.94,
9136.046666666667, 9523.383333333333, 9852.293333333333, 9176.686666666666,
9063.51, 9730.656666666666, 10091.246666666666, 9488.433333333332,
9463.633333333333, 9551.72, 9225.066666666668, 9327.333333333334,
9469.856666666667, 9415.986666666666, 9649.613333333333, 9387.823333333334,
8895.91, 9685.313333333334, 9359.056666666667, 9284.94, 9378.596666666666,
9449.38, 9552.533333333333, 9438.363333333333, 9167.5, 9514.01,
9595.913333333334, 9465.586666666666, 9669.53, 9041.406666666666, 9546.93,
9159.92, 9554.966666666667, 9411.73, 9416.24, 9240.42, 8994.78, 9154.67,
9485.893333333333, 9633.103333333333, 9493.89, 9764.93, 9566.633333333333,
9253.54, 9979.93, 9466.87, 9236.43, 9104.596666666666, 9746.566666666668,
9693.486666666666, 9552.893333333333, 9234.35, 9391.93, 8985.356666666667,
9727.69, 9604.243333333334, 9514.083333333334, 8731.003333333334,
9300.353333333333, 9773.823333333334, 9601.393333333333, 9778.63,
9691.953333333333, 9059.74, 9717.616666666667, 9529.56, 9573.136666666667,
9583.483333333334, 9278.98, 9376.92, 9533.306666666667, 9241.59, 8700.09,
9220.096666666666, 9329.66, 9578.323333333334, 9388.613333333333,
9170.606666666667, 9836.57, 9144.41, 9943.16, 9597.176666666666, 9457.77,
9431.29, 9111.416666666666, 9174.113333333333, 9350.636666666667,

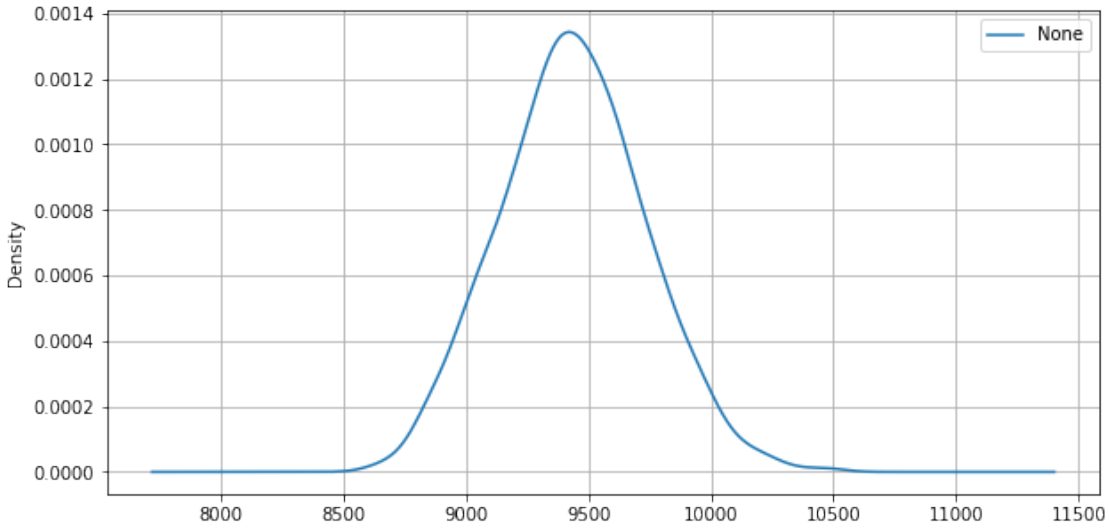
9963.586666666666, 9814.45, 9080.903333333334, 8957.953333333333,
9179.066666666668, 9253.05, 8793.56, 9314.563333333334, 9531.433333333332,
9267.813333333334, 9473.063333333334, 9327.47, 9303.693333333333, 9534.72,
9384.583333333334, 8990.396666666667, 9727.393333333333, 9429.846666666666,
9491.41, 9805.76, 8729.703333333333, 10032.833333333334, 9514.92,
9142.106666666667, 9611.913333333334, 9409.456666666667, 9141.18,
9185.673333333334, 9173.183333333332, 9839.493333333334, 9620.316666666668,
9150.26, 9730.6, 9551.18, 9590.16, 9528.706666666667, 9680.7, 9634.58,
9508.563333333334, 9791.72, 9680.136666666667, 9233.0, 9721.206666666667,
9478.376666666667, 9265.953333333333, 9517.613333333333, 9558.703333333333,
9629.616666666667, 9616.286666666667, 9738.9, 9461.59, 9984.726666666667,
9490.03, 9596.513333333334, 9384.513333333334, 9279.54, 9172.36, 9418.76,
9214.606666666667, 9592.913333333334, 9894.13, 9363.56, 9448.826666666666,
9394.563333333334, 9483.563333333334, 9749.656666666666, 9755.056666666667,
9261.756666666666, 9352.76, 9925.62, 9497.76, 9090.416666666666,
9510.153333333334, 9297.103333333333, 9104.853333333333, 9626.003333333334,
9689.99, 9360.556666666667, 8996.726666666667, 9693.516666666666,
9631.203333333333, 9091.646666666667, 9478.006666666666, 9320.873333333333,
9622.806666666667, 9524.726666666667, 9548.526666666667, 9511.146666666667,
9327.463333333333, 9628.663333333334, 9699.69, 9172.463333333333, 9565.97,
8931.166666666666, 9293.88, 8995.406666666666, 9633.623333333333,
9423.956666666667, 9427.45, 9603.096666666666, 9821.513333333334,
9462.326666666666, 9559.596666666666, 9331.556666666667, 9061.76, 8647.55,
9243.97, 10090.396666666667, 8900.053333333333, 9474.02, 9204.81,
9562.313333333334, 9498.713333333333, 9533.606666666667, 9768.37, 9432.32,
9257.086666666666, 9519.236666666666, 9672.11, 9586.98, 9708.236666666666,
9655.68, 10242.716666666667, 9152.633333333333, 9736.82, 8957.103333333333,
9141.15, 9915.76, 9667.156666666666, 9100.12, 9002.48, 9357.87,
9815.216666666667, 9551.98, 9704.466666666667, 9722.11, 9437.65, 9172.0,
9444.693333333333, 9507.463333333333, 9473.633333333333, 8876.256666666666,
10090.14, 8944.216666666667, 10429.003333333334, 9143.663333333334, 9527.38,
9295.59, 9619.093333333334, 9824.633333333333, 9707.063333333334, 9105.3,
9936.886666666667, 9059.31, 9459.316666666668, 9187.013333333334, 8916.32,
9421.32, 9376.076666666666, 9421.086666666666, 9049.426666666666,
9210.656666666666, 9161.266666666666, 9302.72, 9542.66, 8750.353333333333,
9568.763333333334, 9417.97, 9485.933333333332, 9095.54, 9627.436666666666,
9256.986666666666, 9335.84, 10012.01, 9369.05, 9858.406666666666,
9872.306666666667, 9672.146666666667, 9010.85, 9236.66, 9829.81, 9285.82,
9753.453333333333, 9089.963333333333, 9296.53, 9828.79, 9445.116666666667,
9559.2, 8768.29, 9471.306666666667, 9531.58, 9713.626666666667,
9079.266666666666, 9215.226666666667, 9852.613333333333, 9618.8, 9105.26,
9465.253333333334, 9952.616666666667, 9636.62, 9014.263333333334, 9661.25,
9334.133333333333, 9419.586666666666, 9299.273333333333, 9311.45,
9487.216666666667, 9433.603333333333, 9447.186666666666, 9840.306666666667,
9476.103333333333, 9481.62, 9398.686666666666, 9228.94, 9621.833333333334,
9393.406666666666, 9374.193333333333, 9474.016666666666, 9453.11, 9215.17,
10037.386666666667, 9417.046666666667, 9227.466666666667, 9517.48, 9503.96,
9164.38, 9766.853333333333, 9275.253333333334, 9069.613333333333,

9362.833333333334, 9350.823333333334, 9327.193333333333, 9592.296666666667,
9564.896666666667, 8953.963333333333, 9194.43, 9662.233333333334,
8923.043333333333, 9323.446666666667, 9392.303333333333, 10074.09, 9056.92,
9227.646666666667, 9318.38, 9546.823333333334, 9033.65, 9644.856666666667,
9447.116666666667, 9439.696666666667, 9211.04, 9469.243333333334,
9411.513333333334, 9262.553333333333, 9285.493333333334, 9671.736666666666,
9128.056666666667, 9188.156666666666, 9839.913333333334, 9426.273333333333,
9362.86, 9634.363333333333, 9811.53, 9340.803333333333, 9355.103333333333,
8749.053333333333, 9438.13, 9567.903333333334, 9397.18, 9307.206666666667,
9938.593333333334, 9220.793333333333, 9513.546666666667, 9249.72,
9173.613333333333, 9379.283333333333, 9501.496666666666, 9689.063333333334,
9485.173333333334, 9826.366666666667, 9658.65, 9049.616666666667,
9866.726666666667, 9476.06, 9647.883333333333, 9439.843333333334, 9268.75,
9811.063333333334, 10063.973333333333, 9877.923333333334, 9045.46, 8878.75,
9393.923333333334, 8704.833333333334, 9423.99, 9259.85, 9463.923333333334,
9222.023333333333, 8722.393333333333, 9716.86, 8869.476666666667, 9325.14,
9270.343333333334, 9421.996666666666, 9990.416666666666, 9473.9,
9080.173333333334, 9646.486666666666, 9326.556666666667, 9378.09,
9560.083333333334, 8755.443333333333, 9616.583333333334, 9651.263333333334,
9153.766666666666, 9363.133333333333, 8947.023333333333, 9548.343333333334,
9157.123333333333, 8959.026666666667, 9573.123333333333, 8994.866666666667,
9529.256666666666, 9547.33, 9197.143333333333, 9269.06, 9042.38,
8904.066666666668, 9145.123333333333, 9312.64, 9664.72, 9327.383333333333,
9430.916666666666, 9230.26, 9443.023333333333, 9175.963333333333, 9885.35,
9243.37, 9450.183333333332, 9281.656666666666, 9486.473333333333,
9553.873333333333, 9332.243333333334, 9282.21, 9227.523333333333,
9318.456666666667, 9470.77, 9034.97, 9330.333333333334, 10005.406666666666,
9476.096666666666, 9694.406666666666, 9568.82, 9228.223333333333,
9009.943333333333, 9293.383333333333, 9754.17, 9315.216666666667,
9442.473333333333, 9448.043333333333, 10239.743333333334, 9528.356666666667,
8987.106666666667, 9639.993333333334, 9402.503333333334, 9590.266666666666,
8974.53, 9516.79, 8971.733333333334, 9100.823333333334, 9320.236666666666,
9444.9, 9346.31, 9309.903333333334, 9011.833333333334, 9908.21,
9911.593333333334, 8691.49, 9531.41, 9482.27, 9884.716666666667,
9412.376666666667, 9821.973333333333, 8926.406666666666, 9724.57,
9456.716666666667, 9616.16, 9244.27, 9451.31, 8910.096666666666,
10044.606666666667, 9261.056666666667, 9625.793333333333, 8725.56, 9171.7,
9173.36, 9424.253333333334, 9394.496666666666, 9554.926666666666,
9548.393333333333, 9142.086666666666, 9351.526666666667, 9753.19,
9317.026666666667, 8640.2, 9373.29, 9489.906666666666, 9556.813333333334,
9435.95, 9096.316666666668, 9562.46, 9183.923333333334, 9461.896666666667,
9465.623333333333, 9894.71, 9445.1, 9646.333333333334, 9280.733333333334,
9515.11, 9318.92, 9510.736666666666, 9578.883333333333, 8999.63,
9749.226666666667, 8950.97, 9134.316666666668, 9335.016666666666,
9746.013333333334, 9515.906666666666, 9547.583333333334, 9249.703333333333,
9305.346666666666, 10010.346666666666, 9119.396666666667, 9637.133333333333,
9736.583333333334, 9177.426666666666, 9663.276666666667, 9228.523333333333,
9031.436666666666, 9575.056666666667, 9559.613333333333, 9403.246666666666,

```
8847.246666666666, 9403.44, 9284.65, 9417.56, 9473.056666666667,
9502.713333333333, 9386.66, 9168.846666666666, 9243.413333333334,
8947.333333333334, 9116.186666666666, 9754.74, 9914.983333333334, 9392.4,
9595.56, 9084.943333333333, 9106.253333333334, 8958.393333333333, 9218.08,
9212.75, 9005.626666666667, 9303.18, 9131.973333333333, 9810.923333333334,
9325.896666666667, 9289.476666666667, 9401.896666666667, 9390.4,
9488.536666666667, 9721.48, 9492.91, 9353.076666666666, 9165.803333333333,
9578.846666666666, 9389.703333333333, 9742.963333333333, 9594.96, 9514.04,
9434.426666666666, 9099.433333333332, 9313.286666666667, 9295.11, 9626.77,
10110.166666666666, 9609.77, 9492.266666666666, 9278.866666666667,
9420.186666666666, 9464.116666666667, 9680.04, 9299.566666666668,
9311.493333333334, 9648.1, 9572.56, 9533.36, 9284.963333333333,
9584.113333333333, 9777.663333333334, 9303.686666666666, 9165.643333333333,
9802.38, 9434.336666666666, 8946.4, 9353.3, 9501.86, 9401.863333333333, 9759.53,
9763.66, 9401.32, 9798.396666666667, 9217.766666666666, 9873.123333333333,
9581.01, 8889.04, 9893.496666666666, 8891.47, 9689.726666666667, 9676.5,
9285.553333333333, 9769.376666666667, 9789.41, 9385.85, 9340.523333333333,
8984.76, 8826.52, 10034.066666666668, 9495.39, 9649.056666666667,
9612.636666666667, 9730.916666666666, 9255.14, 9327.126666666667, 9617.34,
9325.766666666666, 9222.236666666666, 9737.793333333333, 8872.036666666667,
9791.036666666667, 9226.216666666667, 9650.25, 9380.133333333333,
9803.773333333333, 9277.41, 9842.34, 9552.67, 9390.043333333333,
9312.496666666666, 9448.283333333333, 9733.756666666666, 9466.38,
9263.006666666666, 9327.393333333333, 9907.926666666666, 9058.743333333334,
9128.003333333334, 9486.57, 9027.05, 9620.636666666667]
```

[17]:
```
# finding population mean
population_mean_list=[np.random.choice(d,size=300,replace=True).mean() for i in␣
 ↪range (1000)]
pd.Series(population_mean_list).
 ↪plot(kind="kde",figsize=(10,5),legend=True,grid=True)
```

[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3589f80550>



[17]:

[18]:
```
d=np.array(data[data["Gender"]=="M"].Purchase) #formulate data to analyze
# making simulation function in order to simulate for various sample size and␣
 ↪iteration to observe its effect
def simulation_iter (data,sample_size,iteration):
  sample_mean=np.random.choice(d,size=sample_size,replace=True).mean()
  population_mean_list=[np.random.choice(d,size=sample_size,replace=True).
 ↪mean() for i in range (iteration)]
  return pd.Series(population_mean_list).
 ↪plot(kind="kde",figsize=(20,5),label=iteration,legend=True,grid=True)
```

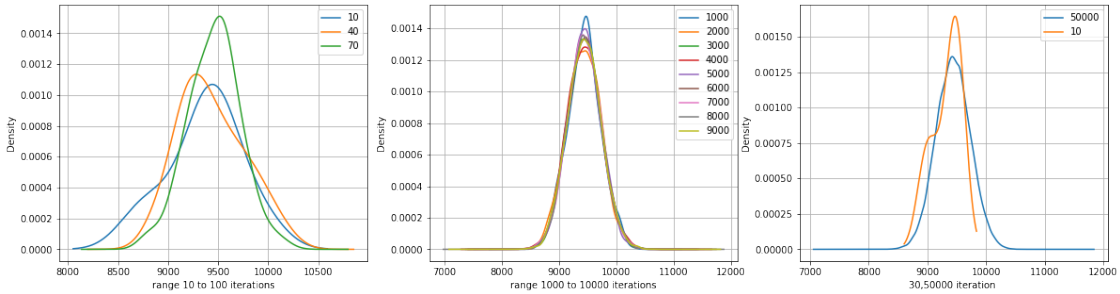
[19]:
```
plt.subplot(1,3,1) #subplot1
for i in range(10,100,30):  #simulation for iterations range 10 to 100
  simulation_iter(d,300,i)
plt.xlabel("range 10 to 100 iterations")

plt.subplot(1,3,2)  #simulation for iteration range 1000 to 10000
for i in range(1000,10000,1000):
  simulation_iter(d,300,i)
plt.xlabel("range 1000 to 10000 iterations")

plt.subplot(1,3,3)  # simulation for 15000 iteration
simulation_iter(d,300,50000)
simulation_iter(d,300,10)


plt.xlabel("30,50000 iteration ")
```

[19]: Text(0.5, 0, '30,50000 iteration ')



As no of times of experiment done is more the distribution becomes closer to **normal distribution**

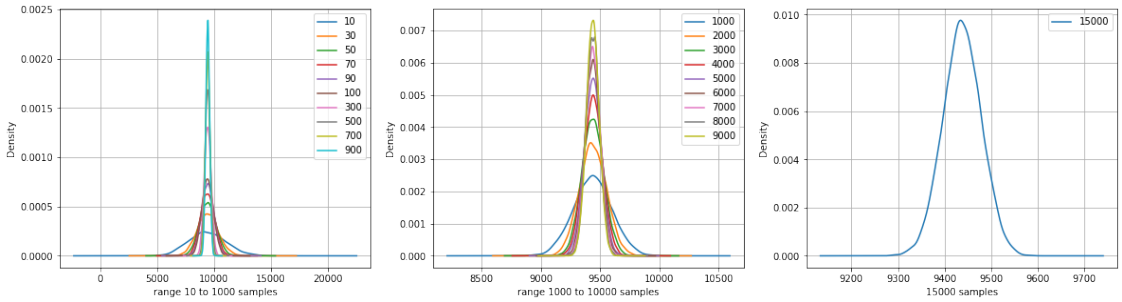
[20]:
```
# making simulation function in order to simulate for various sample size and
 ↪iteration to observe its effect
def simulation_sample (d,sample_size,iteration):
  sample_mean=np.random.choice(d,size=sample_size,replace=True).mean()
  population_mean_list=[np.random.choice(d,size=sample_size,replace=True).
 ↪mean() for i in range (iteration)]
  return pd.Series(population_mean_list).
 ↪plot(kind="kde",figsize=(20,5),label=sample_size,legend=True,grid=True) #
 ↪here only i changed label
```

[21]:
```
plt.subplot(1,3,1) #subplot1
for i in range(10,100,20):  #simulation for frequency range 10 to 1000
  simulation_sample(d,i,10000)
for i in range(100,1000,200):  #simulation for different frequency
  simulation_sample(d,i,10000)
plt.xlabel("range 10 to 1000 samples")

plt.subplot(1,3,2)  #simulation for frequency range 1000 to 10000
for i in range(1000,10000,1000):
  simulation_sample(d,i,10000)
plt.xlabel("range 1000 to 10000 samples")

plt.subplot(1,3,3)  # simulation for 15000 samples
simulation_sample(d,15000,10000)
plt.xlabel("15000 samples")
```

[21]: Text(0.5, 0, '15000 samples')



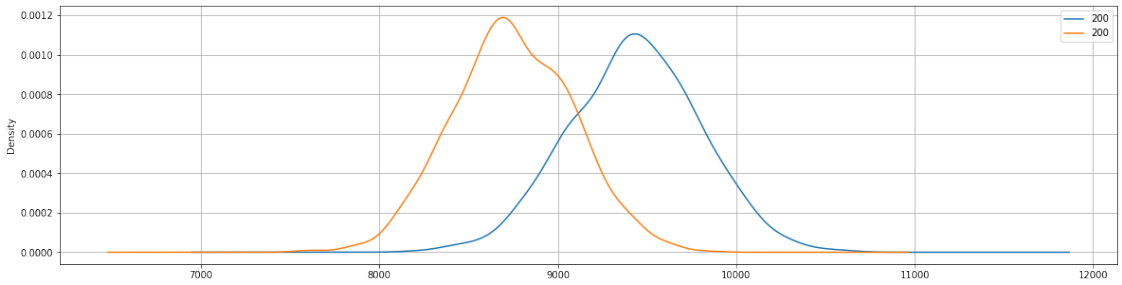
As sample size increases sample mean becomes closer to population mean

from the two experiments we understands we should take more samples and more no of iteration

[22]:
```
M=np.array(data[data["Gender"]=="M"].Purchase)
F=np.array(data[data["Gender"]=="F"].Purchase)
def simulation_sample (df1,sample_size,iteration):
  sample_mean=np.random.choice(df1,size=sample_size,replace=True).mean()
  sample_mean_list=[np.random.choice(df1,size=sample_size,replace=True).mean()␣
 ↪for i in range (iteration)]
  return pd.Series(sample_mean_list).
 ↪plot(kind="kde",figsize=(20,5),label=sample_size,legend=True,grid=True) #␣
 ↪here only i changed label
```

[23]:
```
simulation_sample(M,200,1000)
simulation_sample(F,200,1000)
```

[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f35872e0950>
```



[24]:
```
# finding confidence intervals
def confidence_interval(df1,cf,sample_size=300,iteration=1000):
  sample_mean=np.random.choice(df1,size=sample_size,replace=True).mean()
  sample_mean_list=[np.random.choice(df1,size=sample_size,replace=True).mean()␣
 ↪for i in range(iteration)]
  population_mean=np.array(sample_mean_list).mean()  # finding population mean
```

```
  std_error=np.array(sample_mean_list).std() # finding std error
  lower=population_mean-cf*std_error # finding lower limit
  upper=population_mean+cf*std_error # finding upper limit
  return lower,upper
```

for 95% confidence z score euivalent = 1.96 leave 2.5% from both end

[28]:
```
M=np.array(data[data["Gender"]=="M"].Purchase) # data of male
F=np.array(data[data["Gender"]=="F"].Purchase) # data of female



# finding overlaps

a,b=confidence_interval(M,1.96)
c,d=confidence_interval(F,1.96)
x = range(int(a),int(b))
y = range(int(c),int(d))

def range_overlapping(x, y):
    if x.start == x.stop or y.start == y.stop:
        return False
    return x.start <= y.stop and y.start <= x.stop
print(range_overlapping(x, y))
print(f"confidence interval of m is {a},{b}")
print(f"confidence interval of f is {c},{d}")
```

```
True
confidence interval of m is 8859.755365875615,10002.54316079105
confidence interval of f is 8203.011801404205,9278.401331929126
```

As interval is overlapping we can say there is no difference in purchasing behaviour for male and female

[29]:
```
data.columns
```

[29]:
```
Index(['User_ID', 'Product_ID', 'Gender', 'Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category',
       'Purchase'],
      dtype='object')
```

[30]:
```
data.sample(10)
```

[30]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	\
498633	1004771	P00002942	F	26-35	4	1	
435280	1001051	P00004542	F	26-35	0	1	
206996	1001900	P00249542	M	55+	12	1	
344381	1005011	P00044442	M	18-25	4	2	

```
446332  1002772  P00329542      M  36-45          17               1
468074  1000078  P00065042      F  46-50           1               3
211396  1002629  P00001142      M  26-35          11               3
528145  1003411  P00182742      M  18-25           4               1
63640   1003769  P00086842      M  26-35          15               2
134100  1002703  P00313642      M  36-45          11               3
```

	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
498633	0	0	2	9663
435280	0	1	5	6900
206996	3	1	1	15423
344381	1	0	1	15416
446332	3	1	1	15646
468074	0	1	5	5305
211396	0	1	5	6969
528145	4+	0	1	7690
63640	0	0	14	18317
134100	4+	1	6	20218

[31]:
```
check=['Age', 'Occupation', 'City_Category',
       'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Category']
```

[58]:
```
l=[]
for i in data["Age"].unique():
  c=np.array(data[data["Age"]==i].Purchase) # data of j,i

  print(f"confidence interval of Age {i} is {int(a)},{int(b)}")
  a,b=confidence_interval(c,1.96)

  e1=(int(a),(b))
  l.append(e1)
for i in l:
  for j in l:
    a=i[0]
    b=i[1]
    c=j[0]
    d=j[1]
    x = range(int(a),int(b))
    y = range(int(c),int(d))

    def range_overlapping(x, y):
        if x.start == x.stop or y.start == y.stop:
            return False
        return x.start <= y.stop and y.start <= x.stop

    if range_overlapping(x, y) == False:
      print(i,j)
```

```
confidence interval of Age 0-17 is 8676,9848
confidence interval of Age 55+ is 8366,9504
confidence interval of Age 26-35 is 8777,9894
confidence interval of Age 46-50 is 8688,9822
confidence interval of Age 51-55 is 8626,9770
confidence interval of Age 36-45 is 8965,10110
confidence interval of Age 18-25 is 8751,9894
```

no false condition satisfied hence

purchasing behaviour across all ages are overlapping

[57]:
```
l=[]
for i in data["Occupation"].unique():
  c=np.array(data[data["Occupation"]==i].Purchase) # data of j,i

  print(f"confidence interval of Occupation {i} is {int(a)},{int(b)}")
  a,b=confidence_interval(c,1.96)

  e1=(int(a),(b))
  l.append(e1)
for i in l:
  for j in l:
    a=i[0]
    b=i[1]
    c=j[0]
    d=j[1]
    x = range(int(a),int(b))
    y = range(int(c),int(d))

    def range_overlapping(x, y):
        if x.start == x.stop or y.start == y.stop:
            return False
        return x.start <= y.stop and y.start <= x.stop


    if range_overlapping(x, y) == False:
      print(i,j)
```

```
confidence interval of Occupation 10 is 8700,9796
confidence interval of Occupation 16 is 8361,9552
confidence interval of Occupation 15 is 8830,9943
confidence interval of Occupation 7 is 9194,10364
confidence interval of Occupation 20 is 8834,9999
confidence interval of Occupation 9 is 8275,9387
confidence interval of Occupation 1 is 8133,9140
confidence interval of Occupation 12 is 8434,9502
confidence interval of Occupation 17 is 9219,10352
confidence interval of Occupation 0 is 9223,10414
```

```
confidence interval of Occupation 3 is 8583,9656
confidence interval of Occupation 4 is 8616,9746
confidence interval of Occupation 11 is 8629,9765
confidence interval of Occupation 8 is 8636,9780
confidence interval of Occupation 19 is 8982,10101
confidence interval of Occupation 2 is 8113,9296
confidence interval of Occupation 18 is 8403,9465
confidence interval of Occupation 5 is 8591,9756
confidence interval of Occupation 14 is 8757,9915
confidence interval of Occupation 13 is 8931,10092
confidence interval of Occupation 6 is 8727,9870
(9194, 10364.723430474534) (8133, 9140.757284028427)
(8133, 9140.757284028427) (9194, 10364.723430474534)
(8133, 9140.757284028427) (9219, 10352.965978064383)
(8133, 9140.757284028427) (9223, 10414.18540512788)
(9219, 10352.965978064383) (8133, 9140.757284028427)
(9223, 10414.18540512788) (8133, 9140.757284028427)
```

occupation 7 and coccupation 1 are not overlapping

occupation 1 and coccupation 17 are not overlapping

occupation 1 and coccupation 0 are not overlapping

[59]:
```
l=[]
for i in data["City_Category"].unique():
  c=np.array(data[data["City_Category"]==i].Purchase) # data of j,i

  print(f"confidence interval of City_Category {i} is {int(a)},{int(b)}")
  a,b=confidence_interval(c,1.96)

  e1=(int(a),(b))
  l.append(e1)
for i in l:
  for j in l:
    a=i[0]
    b=i[1]
    c=j[0]
    d=j[1]
    x = range(int(a),int(b))
    y = range(int(c),int(d))

    def range_overlapping(x, y):
        if x.start == x.stop or y.start == y.stop:
            return False
        return x.start <= y.stop and y.start <= x.stop


    if range_overlapping(x, y) == False:
```

```
      print(i,j)
```

```
confidence interval of City_Category 1 is 8600,9749
confidence interval of City_Category 3 is 8350,9455
confidence interval of City_Category 2 is 9135,10312
```

all across cities purchasing behaviour is same

[60]:
```
l=[]
for i in data["Marital_Status"].unique():
  c=np.array(data[data["Marital_Status"]==i].Purchase) # data of j,i

  print(f"confidence interval of Marital_Status {i} is {int(a)},{int(b)}")
  a,b=confidence_interval(c,1.96)

  e1=(int(a),(b))
  l.append(e1)
for i in l:
  for j in l:
    a=i[0]
    b=i[1]
    c=j[0]
    d=j[1]
    x = range(int(a),int(b))
    y = range(int(c),int(d))

    def range_overlapping(x, y):
        if x.start == x.stop or y.start == y.stop:
            return False
        return x.start <= y.stop and y.start <= x.stop


    if range_overlapping(x, y) == False:
      print(i,j)
```

```
confidence interval of Marital_Status 0 is 8595,9712
confidence interval of Marital_Status 1 is 8724,9797
```

Purchasing behaviour is not affected by Marital_Status

[63]:
```
l=[]
for i in data["Product_Category"].unique():
  c=np.array(data[data["Product_Category"]==i].Purchase) # data of j,i

  print(f"confidence interval of Product_Category {i} is {int(a)},{int(b)}")
  a,b=confidence_interval(c,1.96)

  e1=(int(a),(b))
  l.append(e1)
```

```
for i in l:
  for j in l:
    a=i[0]
    b=i[1]
    c=j[0]
    d=j[1]
    x = range(int(a),int(b))
    y = range(int(c),int(d))

    def range_overlapping(x, y):
        if x.start == x.stop or y.start == y.stop:
            return False
        return x.start <= y.stop and y.start <= x.stop


    if range_overlapping(x, y) == False:
      print(i,j)
```

```
confidence interval of Product_Category 3 is 35,38
confidence interval of Product_Category 1 is 9788,10399
confidence interval of Product_Category 12 is 13146,14090
confidence interval of Product_Category 8 is 1310,1393
confidence interval of Product_Category 5 is 7276,7720
confidence interval of Product_Category 4 is 6028,6449
confidence interval of Product_Category 2 is 2238,2416
confidence interval of Product_Category 6 is 10850,11662
confidence interval of Product_Category 14 is 15383,16280
confidence interval of Product_Category 11 is 12678,13590
confidence interval of Product_Category 13 is 4472,4889
confidence interval of Product_Category 15 is 701,743
confidence interval of Product_Category 7 is 14169,15365
confidence interval of Product_Category 16 is 15909,16834
confidence interval of Product_Category 18 is 14289,15277
confidence interval of Product_Category 10 is 2887,3056
confidence interval of Product_Category 17 is 19190,20154
confidence interval of Product_Category 9 is 9906,10427
confidence interval of Product_Category 20 is 14911,16146
confidence interval of Product_Category 19 is 352,388
(9788, 10399.33600765508) (13146, 14090.698315422394)
(9788, 10399.33600765508) (1310, 1393.3004329655685)
(9788, 10399.33600765508) (7276, 7720.983576369786)
(9788, 10399.33600765508) (6028, 6449.226380188689)
(9788, 10399.33600765508) (2238, 2416.9197539402544)
(9788, 10399.33600765508) (10850, 11662.543405851085)
(9788, 10399.33600765508) (15383, 16280.971502787657)
(9788, 10399.33600765508) (12678, 13590.612771444501)
(9788, 10399.33600765508) (4472, 4889.51839097212)
```

(9788, 10399.33600765508) (701, 743.5944592536625)
(9788, 10399.33600765508) (14169, 15365.774461331614)
(9788, 10399.33600765508) (15909, 16834.83887849887)
(9788, 10399.33600765508) (14289, 15277.149567511085)
(9788, 10399.33600765508) (2887, 3056.041474420083)
(9788, 10399.33600765508) (19190, 20154.628215938883)
(9788, 10399.33600765508) (14911, 16146.364310617817)
(9788, 10399.33600765508) (352, 388.9754777934902)
(9788, 10399.33600765508) (35, 39.02634767902216)
(13146, 14090.698315422394) (9788, 10399.33600765508)
(13146, 14090.698315422394) (1310, 1393.3004329655685)
(13146, 14090.698315422394) (7276, 7720.983576369786)
(13146, 14090.698315422394) (6028, 6449.226380188689)
(13146, 14090.698315422394) (2238, 2416.9197539402544)
(13146, 14090.698315422394) (10850, 11662.543405851085)
(13146, 14090.698315422394) (15383, 16280.971502787657)
(13146, 14090.698315422394) (4472, 4889.51839097212)
(13146, 14090.698315422394) (701, 743.5944592536625)
(13146, 14090.698315422394) (14169, 15365.774461331614)
(13146, 14090.698315422394) (15909, 16834.83887849887)
(13146, 14090.698315422394) (14289, 15277.149567511085)
(13146, 14090.698315422394) (2887, 3056.041474420083)
(13146, 14090.698315422394) (19190, 20154.628215938883)
(13146, 14090.698315422394) (9906, 10427.140562523606)
(13146, 14090.698315422394) (14911, 16146.364310617817)
(13146, 14090.698315422394) (352, 388.9754777934902)
(13146, 14090.698315422394) (35, 39.02634767902216)
(1310, 1393.3004329655685) (9788, 10399.33600765508)
(1310, 1393.3004329655685) (13146, 14090.698315422394)
(1310, 1393.3004329655685) (7276, 7720.983576369786)
(1310, 1393.3004329655685) (6028, 6449.226380188689)
(1310, 1393.3004329655685) (2238, 2416.9197539402544)
(1310, 1393.3004329655685) (10850, 11662.543405851085)
(1310, 1393.3004329655685) (15383, 16280.971502787657)
(1310, 1393.3004329655685) (12678, 13590.612771444501)
(1310, 1393.3004329655685) (4472, 4889.51839097212)
(1310, 1393.3004329655685) (701, 743.5944592536625)
(1310, 1393.3004329655685) (14169, 15365.774461331614)
(1310, 1393.3004329655685) (15909, 16834.83887849887)
(1310, 1393.3004329655685) (14289, 15277.149567511085)
(1310, 1393.3004329655685) (2887, 3056.041474420083)
(1310, 1393.3004329655685) (19190, 20154.628215938883)
(1310, 1393.3004329655685) (9906, 10427.140562523606)
(1310, 1393.3004329655685) (14911, 16146.364310617817)
(1310, 1393.3004329655685) (352, 388.9754777934902)
(1310, 1393.3004329655685) (35, 39.02634767902216)
(7276, 7720.983576369786) (9788, 10399.33600765508)
(7276, 7720.983576369786) (13146, 14090.698315422394)

(7276, 7720.983576369786) (1310, 1393.3004329655685)
(7276, 7720.983576369786) (6028, 6449.226380188689)
(7276, 7720.983576369786) (2238, 2416.9197539402544)
(7276, 7720.983576369786) (10850, 11662.543405851085)
(7276, 7720.983576369786) (15383, 16280.971502787657)
(7276, 7720.983576369786) (12678, 13590.612771444501)
(7276, 7720.983576369786) (4472, 4889.51839097212)
(7276, 7720.983576369786) (701, 743.5944592536625)
(7276, 7720.983576369786) (14169, 15365.774461331614)
(7276, 7720.983576369786) (15909, 16834.83887849887)
(7276, 7720.983576369786) (14289, 15277.149567511085)
(7276, 7720.983576369786) (2887, 3056.041474420083)
(7276, 7720.983576369786) (19190, 20154.628215938883)
(7276, 7720.983576369786) (9906, 10427.140562523606)
(7276, 7720.983576369786) (14911, 16146.364310617817)
(7276, 7720.983576369786) (352, 388.9754777934902)
(7276, 7720.983576369786) (35, 39.02634767902216)
(6028, 6449.226380188689) (9788, 10399.33600765508)
(6028, 6449.226380188689) (13146, 14090.698315422394)
(6028, 6449.226380188689) (1310, 1393.3004329655685)
(6028, 6449.226380188689) (7276, 7720.983576369786)
(6028, 6449.226380188689) (2238, 2416.9197539402544)
(6028, 6449.226380188689) (10850, 11662.543405851085)
(6028, 6449.226380188689) (15383, 16280.971502787657)
(6028, 6449.226380188689) (12678, 13590.612771444501)
(6028, 6449.226380188689) (4472, 4889.51839097212)
(6028, 6449.226380188689) (701, 743.5944592536625)
(6028, 6449.226380188689) (14169, 15365.774461331614)
(6028, 6449.226380188689) (15909, 16834.83887849887)
(6028, 6449.226380188689) (14289, 15277.149567511085)
(6028, 6449.226380188689) (2887, 3056.041474420083)
(6028, 6449.226380188689) (19190, 20154.628215938883)
(6028, 6449.226380188689) (9906, 10427.140562523606)
(6028, 6449.226380188689) (14911, 16146.364310617817)
(6028, 6449.226380188689) (352, 388.9754777934902)
(6028, 6449.226380188689) (35, 39.02634767902216)
(2238, 2416.9197539402544) (9788, 10399.33600765508)
(2238, 2416.9197539402544) (13146, 14090.698315422394)
(2238, 2416.9197539402544) (1310, 1393.3004329655685)
(2238, 2416.9197539402544) (7276, 7720.983576369786)
(2238, 2416.9197539402544) (6028, 6449.226380188689)
(2238, 2416.9197539402544) (10850, 11662.543405851085)
(2238, 2416.9197539402544) (15383, 16280.971502787657)
(2238, 2416.9197539402544) (12678, 13590.612771444501)
(2238, 2416.9197539402544) (4472, 4889.51839097212)
(2238, 2416.9197539402544) (701, 743.5944592536625)
(2238, 2416.9197539402544) (14169, 15365.774461331614)
(2238, 2416.9197539402544) (15909, 16834.83887849887)

(2238, 2416.9197539402544) (14289, 15277.149567511085)
(2238, 2416.9197539402544) (2887, 3056.041474420083)
(2238, 2416.9197539402544) (19190, 20154.628215938883)
(2238, 2416.9197539402544) (9906, 10427.140562523606)
(2238, 2416.9197539402544) (14911, 16146.364310617817)
(2238, 2416.9197539402544) (352, 388.9754777934902)
(2238, 2416.9197539402544) (35, 39.02634767902216)
(10850, 11662.543405851085) (9788, 10399.33600765508)
(10850, 11662.543405851085) (13146, 14090.698315422394)
(10850, 11662.543405851085) (1310, 1393.3004329655685)
(10850, 11662.543405851085) (7276, 7720.983576369786)
(10850, 11662.543405851085) (6028, 6449.226380188689)
(10850, 11662.543405851085) (2238, 2416.9197539402544)
(10850, 11662.543405851085) (15383, 16280.971502787657)
(10850, 11662.543405851085) (12678, 13590.612771444501)
(10850, 11662.543405851085) (4472, 4889.51839097212)
(10850, 11662.543405851085) (701, 743.5944592536625)
(10850, 11662.543405851085) (14169, 15365.774461331614)
(10850, 11662.543405851085) (15909, 16834.83887849887)
(10850, 11662.543405851085) (14289, 15277.149567511085)
(10850, 11662.543405851085) (2887, 3056.041474420083)
(10850, 11662.543405851085) (19190, 20154.628215938883)
(10850, 11662.543405851085) (9906, 10427.140562523606)
(10850, 11662.543405851085) (14911, 16146.364310617817)
(10850, 11662.543405851085) (352, 388.9754777934902)
(10850, 11662.543405851085) (35, 39.02634767902216)
(15383, 16280.971502787657) (9788, 10399.33600765508)
(15383, 16280.971502787657) (13146, 14090.698315422394)
(15383, 16280.971502787657) (1310, 1393.3004329655685)
(15383, 16280.971502787657) (7276, 7720.983576369786)
(15383, 16280.971502787657) (6028, 6449.226380188689)
(15383, 16280.971502787657) (2238, 2416.9197539402544)
(15383, 16280.971502787657) (10850, 11662.543405851085)
(15383, 16280.971502787657) (12678, 13590.612771444501)
(15383, 16280.971502787657) (4472, 4889.51839097212)
(15383, 16280.971502787657) (701, 743.5944592536625)
(15383, 16280.971502787657) (14169, 15365.774461331614)
(15383, 16280.971502787657) (14289, 15277.149567511085)
(15383, 16280.971502787657) (2887, 3056.041474420083)
(15383, 16280.971502787657) (19190, 20154.628215938883)
(15383, 16280.971502787657) (9906, 10427.140562523606)
(15383, 16280.971502787657) (352, 388.9754777934902)
(15383, 16280.971502787657) (35, 39.02634767902216)
(12678, 13590.612771444501) (9788, 10399.33600765508)
(12678, 13590.612771444501) (1310, 1393.3004329655685)
(12678, 13590.612771444501) (7276, 7720.983576369786)
(12678, 13590.612771444501) (6028, 6449.226380188689)
(12678, 13590.612771444501) (2238, 2416.9197539402544)

(12678, 13590.612771444501) (10850, 11662.543405851085)
(12678, 13590.612771444501) (15383, 16280.971502787657)
(12678, 13590.612771444501) (4472, 4889.51839097212)
(12678, 13590.612771444501) (701, 743.5944592536625)
(12678, 13590.612771444501) (14169, 15365.774461331614)
(12678, 13590.612771444501) (15909, 16834.83887849887)
(12678, 13590.612771444501) (14289, 15277.149567511085)
(12678, 13590.612771444501) (2887, 3056.041474420083)
(12678, 13590.612771444501) (19190, 20154.628215938883)
(12678, 13590.612771444501) (9906, 10427.140562523606)
(12678, 13590.612771444501) (14911, 16146.364310617817)
(12678, 13590.612771444501) (352, 388.9754777934902)
(12678, 13590.612771444501) (35, 39.02634767902216)
(4472, 4889.51839097212) (9788, 10399.33600765508)
(4472, 4889.51839097212) (13146, 14090.698315422394)
(4472, 4889.51839097212) (1310, 1393.3004329655685)
(4472, 4889.51839097212) (7276, 7720.983576369786)
(4472, 4889.51839097212) (6028, 6449.226380188689)
(4472, 4889.51839097212) (2238, 2416.9197539402544)
(4472, 4889.51839097212) (10850, 11662.543405851085)
(4472, 4889.51839097212) (15383, 16280.971502787657)
(4472, 4889.51839097212) (12678, 13590.612771444501)
(4472, 4889.51839097212) (701, 743.5944592536625)
(4472, 4889.51839097212) (14169, 15365.774461331614)
(4472, 4889.51839097212) (15909, 16834.83887849887)
(4472, 4889.51839097212) (14289, 15277.149567511085)
(4472, 4889.51839097212) (2887, 3056.041474420083)
(4472, 4889.51839097212) (19190, 20154.628215938883)
(4472, 4889.51839097212) (9906, 10427.140562523606)
(4472, 4889.51839097212) (14911, 16146.364310617817)
(4472, 4889.51839097212) (352, 388.9754777934902)
(4472, 4889.51839097212) (35, 39.02634767902216)
(701, 743.5944592536625) (9788, 10399.33600765508)
(701, 743.5944592536625) (13146, 14090.698315422394)
(701, 743.5944592536625) (1310, 1393.3004329655685)
(701, 743.5944592536625) (7276, 7720.983576369786)
(701, 743.5944592536625) (6028, 6449.226380188689)
(701, 743.5944592536625) (2238, 2416.9197539402544)
(701, 743.5944592536625) (10850, 11662.543405851085)
(701, 743.5944592536625) (15383, 16280.971502787657)
(701, 743.5944592536625) (12678, 13590.612771444501)
(701, 743.5944592536625) (4472, 4889.51839097212)
(701, 743.5944592536625) (14169, 15365.774461331614)
(701, 743.5944592536625) (15909, 16834.83887849887)
(701, 743.5944592536625) (14289, 15277.149567511085)
(701, 743.5944592536625) (2887, 3056.041474420083)
(701, 743.5944592536625) (19190, 20154.628215938883)
(701, 743.5944592536625) (9906, 10427.140562523606)

(701, 743.5944592536625) (14911, 16146.364310617817)
(701, 743.5944592536625) (352, 388.9754777934902)
(701, 743.5944592536625) (35, 39.02634767902216)
(14169, 15365.774461331614) (9788, 10399.33600765508)
(14169, 15365.774461331614) (13146, 14090.698315422394)
(14169, 15365.774461331614) (1310, 1393.3004329655685)
(14169, 15365.774461331614) (7276, 7720.983576369786)
(14169, 15365.774461331614) (6028, 6449.226380188689)
(14169, 15365.774461331614) (2238, 2416.9197539402544)
(14169, 15365.774461331614) (10850, 11662.543405851085)
(14169, 15365.774461331614) (15383, 16280.971502787657)
(14169, 15365.774461331614) (12678, 13590.612771444501)
(14169, 15365.774461331614) (4472, 4889.51839097212)
(14169, 15365.774461331614) (701, 743.5944592536625)
(14169, 15365.774461331614) (15909, 16834.83887849887)
(14169, 15365.774461331614) (2887, 3056.041474420083)
(14169, 15365.774461331614) (19190, 20154.628215938883)
(14169, 15365.774461331614) (9906, 10427.140562523606)
(14169, 15365.774461331614) (352, 388.9754777934902)
(14169, 15365.774461331614) (35, 39.02634767902216)
(15909, 16834.83887849887) (9788, 10399.33600765508)
(15909, 16834.83887849887) (13146, 14090.698315422394)
(15909, 16834.83887849887) (1310, 1393.3004329655685)
(15909, 16834.83887849887) (7276, 7720.983576369786)
(15909, 16834.83887849887) (6028, 6449.226380188689)
(15909, 16834.83887849887) (2238, 2416.9197539402544)
(15909, 16834.83887849887) (10850, 11662.543405851085)
(15909, 16834.83887849887) (12678, 13590.612771444501)
(15909, 16834.83887849887) (4472, 4889.51839097212)
(15909, 16834.83887849887) (701, 743.5944592536625)
(15909, 16834.83887849887) (14169, 15365.774461331614)
(15909, 16834.83887849887) (14289, 15277.149567511085)
(15909, 16834.83887849887) (2887, 3056.041474420083)
(15909, 16834.83887849887) (19190, 20154.628215938883)
(15909, 16834.83887849887) (9906, 10427.140562523606)
(15909, 16834.83887849887) (352, 388.9754777934902)
(15909, 16834.83887849887) (35, 39.02634767902216)
(14289, 15277.149567511085) (9788, 10399.33600765508)
(14289, 15277.149567511085) (13146, 14090.698315422394)
(14289, 15277.149567511085) (1310, 1393.3004329655685)
(14289, 15277.149567511085) (7276, 7720.983576369786)
(14289, 15277.149567511085) (6028, 6449.226380188689)
(14289, 15277.149567511085) (2238, 2416.9197539402544)
(14289, 15277.149567511085) (10850, 11662.543405851085)
(14289, 15277.149567511085) (15383, 16280.971502787657)
(14289, 15277.149567511085) (12678, 13590.612771444501)
(14289, 15277.149567511085) (4472, 4889.51839097212)
(14289, 15277.149567511085) (701, 743.5944592536625)

(14289, 15277.149567511085) (15909, 16834.83887849887)
(14289, 15277.149567511085) (2887, 3056.041474420083)
(14289, 15277.149567511085) (19190, 20154.628215938883)
(14289, 15277.149567511085) (9906, 10427.140562523606)
(14289, 15277.149567511085) (352, 388.9754777934902)
(14289, 15277.149567511085) (35, 39.02634767902216)
(2887, 3056.041474420083) (9788, 10399.33600765508)
(2887, 3056.041474420083) (13146, 14090.698315422394)
(2887, 3056.041474420083) (1310, 1393.3004329655685)
(2887, 3056.041474420083) (7276, 7720.983576369786)
(2887, 3056.041474420083) (6028, 6449.226380188689)
(2887, 3056.041474420083) (2238, 2416.9197539402544)
(2887, 3056.041474420083) (10850, 11662.543405851085)
(2887, 3056.041474420083) (15383, 16280.971502787657)
(2887, 3056.041474420083) (12678, 13590.612771444501)
(2887, 3056.041474420083) (4472, 4889.51839097212)
(2887, 3056.041474420083) (701, 743.5944592536625)
(2887, 3056.041474420083) (14169, 15365.774461331614)
(2887, 3056.041474420083) (15909, 16834.83887849887)
(2887, 3056.041474420083) (14289, 15277.149567511085)
(2887, 3056.041474420083) (19190, 20154.628215938883)
(2887, 3056.041474420083) (9906, 10427.140562523606)
(2887, 3056.041474420083) (14911, 16146.364310617817)
(2887, 3056.041474420083) (352, 388.9754777934902)
(2887, 3056.041474420083) (35, 39.02634767902216)
(19190, 20154.628215938883) (9788, 10399.33600765508)
(19190, 20154.628215938883) (13146, 14090.698315422394)
(19190, 20154.628215938883) (1310, 1393.3004329655685)
(19190, 20154.628215938883) (7276, 7720.983576369786)
(19190, 20154.628215938883) (6028, 6449.226380188689)
(19190, 20154.628215938883) (2238, 2416.9197539402544)
(19190, 20154.628215938883) (10850, 11662.543405851085)
(19190, 20154.628215938883) (15383, 16280.971502787657)
(19190, 20154.628215938883) (12678, 13590.612771444501)
(19190, 20154.628215938883) (4472, 4889.51839097212)
(19190, 20154.628215938883) (701, 743.5944592536625)
(19190, 20154.628215938883) (14169, 15365.774461331614)
(19190, 20154.628215938883) (15909, 16834.83887849887)
(19190, 20154.628215938883) (14289, 15277.149567511085)
(19190, 20154.628215938883) (2887, 3056.041474420083)
(19190, 20154.628215938883) (9906, 10427.140562523606)
(19190, 20154.628215938883) (14911, 16146.364310617817)
(19190, 20154.628215938883) (352, 388.9754777934902)
(19190, 20154.628215938883) (35, 39.02634767902216)
(9906, 10427.140562523606) (13146, 14090.698315422394)
(9906, 10427.140562523606) (1310, 1393.3004329655685)
(9906, 10427.140562523606) (7276, 7720.983576369786)
(9906, 10427.140562523606) (6028, 6449.226380188689)

(9906, 10427.140562523606) (2238, 2416.9197539402544)
(9906, 10427.140562523606) (10850, 11662.543405851085)
(9906, 10427.140562523606) (15383, 16280.971502787657)
(9906, 10427.140562523606) (12678, 13590.612771444501)
(9906, 10427.140562523606) (4472, 4889.51839097212)
(9906, 10427.140562523606) (701, 743.5944592536625)
(9906, 10427.140562523606) (14169, 15365.774461331614)
(9906, 10427.140562523606) (15909, 16834.83887849887)
(9906, 10427.140562523606) (14289, 15277.149567511085)
(9906, 10427.140562523606) (2887, 3056.041474420083)
(9906, 10427.140562523606) (19190, 20154.628215938883)
(9906, 10427.140562523606) (14911, 16146.364310617817)
(9906, 10427.140562523606) (352, 388.9754777934902)
(9906, 10427.140562523606) (35, 39.02634767902216)
(14911, 16146.364310617817) (9788, 10399.33600765508)
(14911, 16146.364310617817) (13146, 14090.698315422394)
(14911, 16146.364310617817) (1310, 1393.3004329655685)
(14911, 16146.364310617817) (7276, 7720.983576369786)
(14911, 16146.364310617817) (6028, 6449.226380188689)
(14911, 16146.364310617817) (2238, 2416.9197539402544)
(14911, 16146.364310617817) (10850, 11662.543405851085)
(14911, 16146.364310617817) (12678, 13590.612771444501)
(14911, 16146.364310617817) (4472, 4889.51839097212)
(14911, 16146.364310617817) (701, 743.5944592536625)
(14911, 16146.364310617817) (2887, 3056.041474420083)
(14911, 16146.364310617817) (19190, 20154.628215938883)
(14911, 16146.364310617817) (9906, 10427.140562523606)
(14911, 16146.364310617817) (352, 388.9754777934902)
(14911, 16146.364310617817) (35, 39.02634767902216)
(352, 388.9754777934902) (9788, 10399.33600765508)
(352, 388.9754777934902) (13146, 14090.698315422394)
(352, 388.9754777934902) (1310, 1393.3004329655685)
(352, 388.9754777934902) (7276, 7720.983576369786)
(352, 388.9754777934902) (6028, 6449.226380188689)
(352, 388.9754777934902) (2238, 2416.9197539402544)
(352, 388.9754777934902) (10850, 11662.543405851085)
(352, 388.9754777934902) (15383, 16280.971502787657)
(352, 388.9754777934902) (12678, 13590.612771444501)
(352, 388.9754777934902) (4472, 4889.51839097212)
(352, 388.9754777934902) (701, 743.5944592536625)
(352, 388.9754777934902) (14169, 15365.774461331614)
(352, 388.9754777934902) (15909, 16834.83887849887)
(352, 388.9754777934902) (14289, 15277.149567511085)
(352, 388.9754777934902) (2887, 3056.041474420083)
(352, 388.9754777934902) (19190, 20154.628215938883)
(352, 388.9754777934902) (9906, 10427.140562523606)
(352, 388.9754777934902) (14911, 16146.364310617817)
(352, 388.9754777934902) (35, 39.02634767902216)

```
(35, 39.02634767902216) (9788, 10399.33600765508)
(35, 39.02634767902216) (13146, 14090.698315422394)
(35, 39.02634767902216) (1310, 1393.3004329655685)
(35, 39.02634767902216) (7276, 7720.983576369786)
(35, 39.02634767902216) (6028, 6449.226380188689)
(35, 39.02634767902216) (2238, 2416.9197539402544)
(35, 39.02634767902216) (10850, 11662.543405851085)
(35, 39.02634767902216) (15383, 16280.971502787657)
(35, 39.02634767902216) (12678, 13590.612771444501)
(35, 39.02634767902216) (4472, 4889.51839097212)
(35, 39.02634767902216) (701, 743.5944592536625)
(35, 39.02634767902216) (14169, 15365.774461331614)
(35, 39.02634767902216) (15909, 16834.83887849887)
(35, 39.02634767902216) (14289, 15277.149567511085)
(35, 39.02634767902216) (2887, 3056.041474420083)
(35, 39.02634767902216) (19190, 20154.628215938883)
(35, 39.02634767902216) (9906, 10427.140562523606)
(35, 39.02634767902216) (14911, 16146.364310617817)
(35, 39.02634767902216) (352, 388.9754777934902)
```

From above simulations we were able to conclude the purchasing behavious across all caterogry is similar except product wise . we should advertise to all cateogory

occupation 7 and coccupation 1 are not overlapping

occupation 1 and coccupation 17 are not overlapping

occupation 1 and coccupation 0 are not overlapping

hence we concluded occupation 1,7,17 and 0 are behaving differently try to figure out why

since purchasing behaviour is similar in all cateogry advertise to all to get maximum profit and figure out which product is selling less focuss on those like product 3 is having mean value less but consider how much percentage profit is there, now can't proceed due to missing data of profit

Purchasing behaviour is different for different product

As interval is overlapping we can say there is no difference in purchasing behaviour for male and female

from the experiments we understands we should take more samples and more no of iteration

[65]:
```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install pypandoc
```

```
Reading package lists… Done
Building dependency tree
Reading state information… Done
pandoc is already the newest version (1.19.2.4~dfsg-1build4).
pandoc set to manually installed.
The following package was automatically installed and is no longer required:
  libnvidia-common-460
```

```
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
Suggested packages:
  fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
  | pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
  python-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
  | libtcltk-ruby texlive-pictures-doc vprerex
The following NEW packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive texlive-base
  texlive-binaries texlive-fonts-recommended texlive-latex-base
  texlive-latex-extra texlive-latex-recommended texlive-pictures
  texlive-plain-generic texlive-xetex tipa
0 upgraded, 47 newly installed, 0 to remove and 20 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
```

```
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.9 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.16 [5,093 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.16 [2,265 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.12 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 rake all
12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-did-you-mean all
1.2.0-2 [9,700 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-minitest all
5.10.3-1 [38.6 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-power-assert all
0.3.0-1 [7,952 B]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-test-unit all
3.2.5-1 [61.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libruby2.5
amd64 2.5.1-1ubuntu1.12 [3,073 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsynctex1
amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexlua52
amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexluajit2
```

```
amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libzzip-0-13
amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 lmodern all 2.004.5-3
[9,631 kB]
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 preview-latex-style
all 11.91-1ubuntu1 [185 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 t1utils amd64 1.41-2
[56.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tex-gyre all
20160520-1 [4,998 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 texlive-
binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
2017.20180305-1 [951 kB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive all
2017.20180305-1 [14.4 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 7s (21.8 MB/s)
Extracting templates from packages: 100%
Preconfiguring packages …
Selecting previously unselected package fonts-droid-fallback.
(Reading database … 155569 files and directories currently installed.)
Preparing to unpack …/00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb …
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) …
Selecting previously unselected package fonts-lato.
Preparing to unpack …/01-fonts-lato_2.0-2_all.deb …
Unpacking fonts-lato (2.0-2) …
Selecting previously unselected package poppler-data.
Preparing to unpack …/02-poppler-data_0.4.8-2_all.deb …
Unpacking poppler-data (0.4.8-2) …
Selecting previously unselected package tex-common.
Preparing to unpack …/03-tex-common_6.09_all.deb …
```

```
Unpacking tex-common (6.09) …
Selecting previously unselected package fonts-lmodern.
Preparing to unpack …/04-fonts-lmodern_2.004.5-3_all.deb …
Unpacking fonts-lmodern (2.004.5-3) …
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack …/05-fonts-noto-mono_20171026-2_all.deb …
Unpacking fonts-noto-mono (20171026-2) …
Selecting previously unselected package fonts-texgyre.
Preparing to unpack …/06-fonts-texgyre_20160520-1_all.deb …
Unpacking fonts-texgyre (20160520-1) …
Selecting previously unselected package javascript-common.
Preparing to unpack …/07-javascript-common_11_all.deb …
Unpacking javascript-common (11) …
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack …/08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb …
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack …/09-libcupsimage2_2.2.7-1ubuntu2.9_amd64.deb …
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.9) …
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack …/10-libijs-0.35_0.35-13_amd64.deb …
Unpacking libijs-0.35:amd64 (0.35-13) …
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack …/11-libjbig2dec0_0.13-6_amd64.deb …
Unpacking libjbig2dec0:amd64 (0.13-6) …
Selecting previously unselected package libgs9-common.
Preparing to unpack …/12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.16_all.deb
…
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.16) …
Selecting previously unselected package libgs9:amd64.
Preparing to unpack …/13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.16_amd64.deb …
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.16) …
Selecting previously unselected package libjs-jquery.
Preparing to unpack …/14-libjs-jquery_3.2.1-1_all.deb …
Unpacking libjs-jquery (3.2.1-1) …
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack …/15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libpotrace0.
Preparing to unpack …/16-libpotrace0_1.14-2_amd64.deb …
Unpacking libpotrace0 (1.14-2) …
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack …/17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package rubygems-integration.
Preparing to unpack …/18-rubygems-integration_1.11_all.deb …
```

```
Unpacking rubygems-integration (1.11) …
Selecting previously unselected package ruby2.5.
Preparing to unpack …/19-ruby2.5_2.5.1-1ubuntu1.12_amd64.deb …
Unpacking ruby2.5 (2.5.1-1ubuntu1.12) …
Selecting previously unselected package ruby.
Preparing to unpack …/20-ruby_1%3a2.5.1_amd64.deb …
Unpacking ruby (1:2.5.1) …
Selecting previously unselected package rake.
Preparing to unpack …/21-rake_12.3.1-1ubuntu0.1_all.deb …
Unpacking rake (12.3.1-1ubuntu0.1) …
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack …/22-ruby-did-you-mean_1.2.0-2_all.deb …
Unpacking ruby-did-you-mean (1.2.0-2) …
Selecting previously unselected package ruby-minitest.
Preparing to unpack …/23-ruby-minitest_5.10.3-1_all.deb …
Unpacking ruby-minitest (5.10.3-1) …
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack …/24-ruby-net-telnet_0.1.1-2_all.deb …
Unpacking ruby-net-telnet (0.1.1-2) …
Selecting previously unselected package ruby-power-assert.
Preparing to unpack …/25-ruby-power-assert_0.3.0-1_all.deb …
Unpacking ruby-power-assert (0.3.0-1) …
Selecting previously unselected package ruby-test-unit.
Preparing to unpack …/26-ruby-test-unit_3.2.5-1_all.deb …
Unpacking ruby-test-unit (3.2.5-1) …
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack …/27-libruby2.5_2.5.1-1ubuntu1.12_amd64.deb …
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.12) …
Selecting previously unselected package libsynctex1:amd64.
Preparing to unpack …/28-libsynctex1_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack …/29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
…
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
…/30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack …/31-libzzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb …
Unpacking libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Selecting previously unselected package lmodern.
Preparing to unpack …/32-lmodern_2.004.5-3_all.deb …
Unpacking lmodern (2.004.5-3) …
Selecting previously unselected package preview-latex-style.
Preparing to unpack …/33-preview-latex-style_11.91-1ubuntu1_all.deb …
```

```
Unpacking preview-latex-style (11.91-1ubuntu1) …
Selecting previously unselected package t1utils.
Preparing to unpack …/34-t1utils_1.41-2_amd64.deb …
Unpacking t1utils (1.41-2) …
Selecting previously unselected package tex-gyre.
Preparing to unpack …/35-tex-gyre_20160520-1_all.deb …
Unpacking tex-gyre (20160520-1) …
Selecting previously unselected package texlive-binaries.
Preparing to unpack …/36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb …
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
Selecting previously unselected package texlive-base.
Preparing to unpack …/37-texlive-base_2017.20180305-1_all.deb …
Unpacking texlive-base (2017.20180305-1) …
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack …/38-texlive-fonts-recommended_2017.20180305-1_all.deb …
Unpacking texlive-fonts-recommended (2017.20180305-1) …
Selecting previously unselected package texlive-latex-base.
Preparing to unpack …/39-texlive-latex-base_2017.20180305-1_all.deb …
Unpacking texlive-latex-base (2017.20180305-1) …
Selecting previously unselected package texlive-latex-recommended.
Preparing to unpack …/40-texlive-latex-recommended_2017.20180305-1_all.deb …
Unpacking texlive-latex-recommended (2017.20180305-1) …
Selecting previously unselected package texlive.
Preparing to unpack …/41-texlive_2017.20180305-1_all.deb …
Unpacking texlive (2017.20180305-1) …
Selecting previously unselected package texlive-pictures.
Preparing to unpack …/42-texlive-pictures_2017.20180305-1_all.deb …
Unpacking texlive-pictures (2017.20180305-1) …
Selecting previously unselected package texlive-latex-extra.
Preparing to unpack …/43-texlive-latex-extra_2017.20180305-2_all.deb …
Unpacking texlive-latex-extra (2017.20180305-2) …
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack …/44-texlive-plain-generic_2017.20180305-2_all.deb …
Unpacking texlive-plain-generic (2017.20180305-2) …
Selecting previously unselected package tipa.
Preparing to unpack …/45-tipa_2%3a1.3-20_all.deb …
Unpacking tipa (2:1.3-20) …
Selecting previously unselected package texlive-xetex.
Preparing to unpack …/46-texlive-xetex_2017.20180305-1_all.deb …
Unpacking texlive-xetex (2017.20180305-1) …
Setting up libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.16) …
Setting up libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libjs-jquery (3.2.1-1) …
Setting up libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-droid-fallback (1:6.0.1r16-1.1) …
Setting up libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) …
```

```
Setting up tex-common (6.09) …
update-language: texlive-base not installed and configured, doing nothing!
Setting up poppler-data (0.4.8-2) …
Setting up tex-gyre (20160520-1) …
Setting up preview-latex-style (11.91-1ubuntu1) …
Setting up fonts-texgyre (20160520-1) …
Setting up fonts-noto-mono (20171026-2) …
Setting up fonts-lato (2.0-2) …
Setting up libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) …
Setting up libcupsimage2:amd64 (2.2.7-1ubuntu2.9) …
Setting up libjbig2dec0:amd64 (0.13-6) …
Setting up ruby-did-you-mean (1.2.0-2) …
Setting up t1utils (1.41-2) …
Setting up ruby-net-telnet (0.1.1-2) …
Setting up libijs-0.35:amd64 (0.35-13) …
Setting up rubygems-integration (1.11) …
Setting up libpotrace0 (1.14-2) …
Setting up javascript-common (11) …
Setting up ruby-minitest (5.10.3-1) …
Setting up libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) …
Setting up libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.16) …
Setting up libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) …
Setting up fonts-lmodern (2.004.5-3) …
Setting up ruby-power-assert (0.3.0-1) …
Setting up texlive-binaries (2017.20170613.44572-8ubuntu0.1) …
update-alternatives: using /usr/bin/xdvi-xaw to provide /usr/bin/xdvi.bin
(xdvi.bin) in auto mode
update-alternatives: using /usr/bin/bibtex.original to provide /usr/bin/bibtex
(bibtex) in auto mode
Setting up texlive-base (2017.20180305-1) …
mktexlsr: Updating /var/lib/texmf/ls-R-TEXLIVEDIST…
mktexlsr: Updating /var/lib/texmf/ls-R-TEXMFMAIN…
mktexlsr: Updating /var/lib/texmf/ls-R…
mktexlsr: Done.
tl-paper: setting paper size for dvips to a4:
/var/lib/texmf/dvips/config/config-paper.ps
tl-paper: setting paper size for dvipdfmx to a4:
/var/lib/texmf/dvipdfmx/dvipdfmx-paper.cfg
tl-paper: setting paper size for xdvi to a4: /var/lib/texmf/xdvi/XDvi-paper
tl-paper: setting paper size for pdftex to a4:
/var/lib/texmf/tex/generic/config/pdftexconfig.tex
Setting up texlive-fonts-recommended (2017.20180305-1) …
Setting up texlive-plain-generic (2017.20180305-2) …
Setting up texlive-latex-base (2017.20180305-1) …
Setting up lmodern (2.004.5-3) …
Setting up texlive-latex-recommended (2017.20180305-1) …
Setting up texlive-pictures (2017.20180305-1) …
Setting up tipa (2:1.3-20) …
```

```
Regenerating '/var/lib/texmf/fmtutil.cnf-DEBIAN'… done.
Regenerating '/var/lib/texmf/fmtutil.cnf-TEXLIVEDIST'… done.
update-fmtutil has updated the following file(s):
        /var/lib/texmf/fmtutil.cnf-DEBIAN
        /var/lib/texmf/fmtutil.cnf-TEXLIVEDIST
If you want to activate the changes in the above file(s),
you should run fmtutil-sys or fmtutil.
Setting up texlive (2017.20180305-1) …
Setting up texlive-latex-extra (2017.20180305-2) …
Setting up texlive-xetex (2017.20180305-1) …
Setting up ruby2.5 (2.5.1-1ubuntu1.12) …
Setting up ruby (1:2.5.1) …
Setting up ruby-test-unit (3.2.5-1) …
Setting up rake (12.3.1-1ubuntu0.1) …
Setting up libruby2.5:amd64 (2.5.1-1ubuntu1.12) …
Processing triggers for mime-support (3.60ubuntu1) …
Processing triggers for libc-bin (2.27-3ubuntu1.5) …
Processing triggers for man-db (2.8.3-2ubuntu0.1) …
Processing triggers for fontconfig (2.12.6-0ubuntu2) …
Processing triggers for tex-common (6.09) …
Running updmap-sys. This may take some time… done.
Running mktexlsr /var/lib/texmf … done.
Building format(s) --all.
        This may take some time… done.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting pypandoc
  Downloading pypandoc-1.8.1-py3-none-any.whl (20 kB)
Installing collected packages: pypandoc
Successfully installed pypandoc-1.8.1
```

[66]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[72]:
```
!jupyter nbconvert --to PDF "walmart.ipynb"
```

```
[NbConvertApp] WARNING | pattern 'walmart.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
```

```
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abort
conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=
--ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True
```

```
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True
--TemplateExporter.exclude_input=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides']
            or a dotted object name that represents the import path for an
            `Exporter` class
    Default: 'html'
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template file to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                    to output to the directory of each notebook.
To recover
                                    previous default behaviour (outputting to the
```

```
current
                                    working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a
copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-
html-slideshow)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb

            which will convert mynotebook.ipynb to the default format (probably
HTML).

            You can specify the export format with `--to`.
            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX
includes
            'base', 'article' and 'report'.  HTML includes 'basic' and 'full'.
You
            can specify the flavor of the format used.
```

```
            > jupyter nbconvert --to html --template basic mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```

[ ]: