

# Utilization Pilot Analysis

## Cleaning and Pre Processing

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#Settting Display Options
sns.set_theme(style="darkgrid")
```

```
In [2]: #Read in dataset
utilization_df = pd.read_csv('Utilization Pilot - Intern Data Points Exercis

#Check correct import with excel
print(utilization_df.info())
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 603 entries, 0 to 602

Data columns (total 37 columns):

#	Column	Non-Null Count	Dtype
0	Account_name	603 non-null	object
1	Edition/Platform Hybrid	603 non-null	object
2	Effective_Contract_Date	603 non-null	object
3	Implementation Service Manager	603 non-null	object
4	Activation Coach	603 non-null	object
5	Project_name	603 non-null	object
6	Service Type	603 non-null	object
7	Activation Experience	603 non-null	object
8	Implementation_Status__c	603 non-null	object
9	Transition_to_CS_Date	603 non-null	object
10	HPP	603 non-null	float64
11	DTV	603 non-null	int64
12	Trans CS Year-Month	603 non-null	object
13	Trans CS Year-Quarter	603 non-null	object
14	Group Simple	603 non-null	object
15	In Pilot Description	603 non-null	object
16	Sept'22 Trans Qty	603 non-null	object
17	Oct'22		
	Trans Qty	603 non-null	object
18	Nov'22		
	Trans Qty	603 non-null	object
19	Dec'22		
	Trans Qty	603 non-null	object
20	Jan'23		
	Trans Qty	603 non-null	object
21	Feb'23		
	Trans Qty	603 non-null	object
22	Mar'23		
	Trans Qty	603 non-null	object
23	Sept'22 Max Commitment	603 non-null	object
24	Oct'22		
	Max Commitment	603 non-null	object
25	Nov'22		
	Max Commitment	603 non-null	object
26	Dec'22		
	Max Commitment	603 non-null	object
27	Jan'23		
	Max Commitment	603 non-null	object
28	Feb'23		
	Max Commitment	603 non-null	object
29	Mar'23		
	Max Commitment	603 non-null	object
30	Sept'22 Utilization	603 non-null	object
31	Oct'22		
	Utilization	603 non-null	object
32	Nov'22		
	Utilization	603 non-null	object
33	Dec'22		
	Utilization	603 non-null	object
34	Jan'23		
	Utilization	603 non-null	object

```

35 Feb'23
Utilization          603 non-null    object
36 Mar'23
Utilization          603 non-null    object
dtypes: float64(1), int64(1), object(35)
memory usage: 174.4+ KB
None

```

```

In [3]: #Check rows and imports
utilization_df.head(3)

```

```

Out[3]:

```

	Account_name	Edition/Platform Hybrid	Effective_Contract_Date	Implementation Service Manager	Activation Coach
0	FAAC Entrance Solutions UK Ltd	Standard (Universal)	9/1/22	Bill Montgomery	Jahana Yesr
1	Joe Taylor Restoration Inc	Standard	9/1/22	PJ Schoeny	Schoe
2	Stride Property Group	Standard (Universal)	9/1/22	Daryll Aldwin Sales	Dia Ch Mant

3 rows x 37 columns

```

In [4]: #Clean column names
char_remov= ['/', ' ', '-', '_', '"', '\n']

for char in char_remov:
    utilization_df.columns = utilization_df.columns.str.replace(char, '')

print(utilization_df.columns)

Index(['Accountname', 'EditionPlatformHybrid', 'EffectiveContractDate',
      'ImplementationServiceManager', 'ActivationCoach', 'Projectname',
      'ServiceType', 'ActivationExperience', 'ImplementationStatusc',
      'TransitiontoCSDate', 'HPP', 'DTV', 'TransCSYearMonth',
      'TransCSYearQuarter', 'GroupSimple', 'InPilotDescription',
      'Sept22TransQty', 'Oct22TransQty', 'Nov22TransQty', 'Dec22TransQty',
      'Jan23TransQty', 'Feb23TransQty', 'Mar23TransQty',
      'Sept22MaxCommitment', 'Oct22MaxCommitment', 'Nov22MaxCommitment',
      'Dec22MaxCommitment', 'Jan23MaxCommitment', 'Feb23MaxCommitment',
      'Mar23MaxCommitment', 'Sept22Utilization', 'Oct22Utilization',
      'Nov22Utilization', 'Dec22Utilization', 'Jan23Utilization',
      'Feb23Utilization', 'Mar23Utilization'],
      dtype='object')

```

## Dataset Updating and Manipulation

```

In [5]: #Creating Pilot Column including both pilot groups

```

```

utilization_df['isPilot'] = 'Control'
utilization_df.loc[(utilization_df['InPilotDescription'] == 'Pilot Group - E
                    (utilization_df['InPilotDescription'] == 'Pilot Gr

#Checking Distribution between groups and quarter
print(utilization_df['isPilot'].value_counts())
print(utilization_df['TransCSYearQuarter'].value_counts())

#Dropping 2022 Q3 & Q2 based on low data
utilization_df = utilization_df[(utilization_df['TransCSYearQuarter'] == '20
print(utilization_df['TransCSYearQuarter'].value_counts())

# Changed order of TransCSYearMonth column for plot display
utilization_df['TransCSYearMonth'] = pd.Categorical(utilization_df['TransCSY
                    ['2022-10', '2022-11', '2022-12', '2023-01',
                    ordered=True)

#Change month names for plot display
utilization_df['TransCSYearMonth'] = utilization_df['TransCSYearMonth'].repl
                    ['2022-10', '2022-11', '2022-12', '2023-01',
                    ['Oct-22', 'Nov-22', 'Dec-22', 'Jan-23',

#Creating quarter based dataframe
utilization_df_q1 = utilization_df[(utilization_df['TransCSYearQuarter'] ==
utilization_df_q4 = utilization_df[(utilization_df['TransCSYearQuarter'] ==

```

```

isPilot
Control    338
Pilot      265
Name: count, dtype: int64
TransCSYearQuarter
2023-1      389
2022-4      206
2022-3         7
2022-2         1
Name: count, dtype: int64
TransCSYearQuarter
2023-1      389
2022-4      206
Name: count, dtype: int64

```

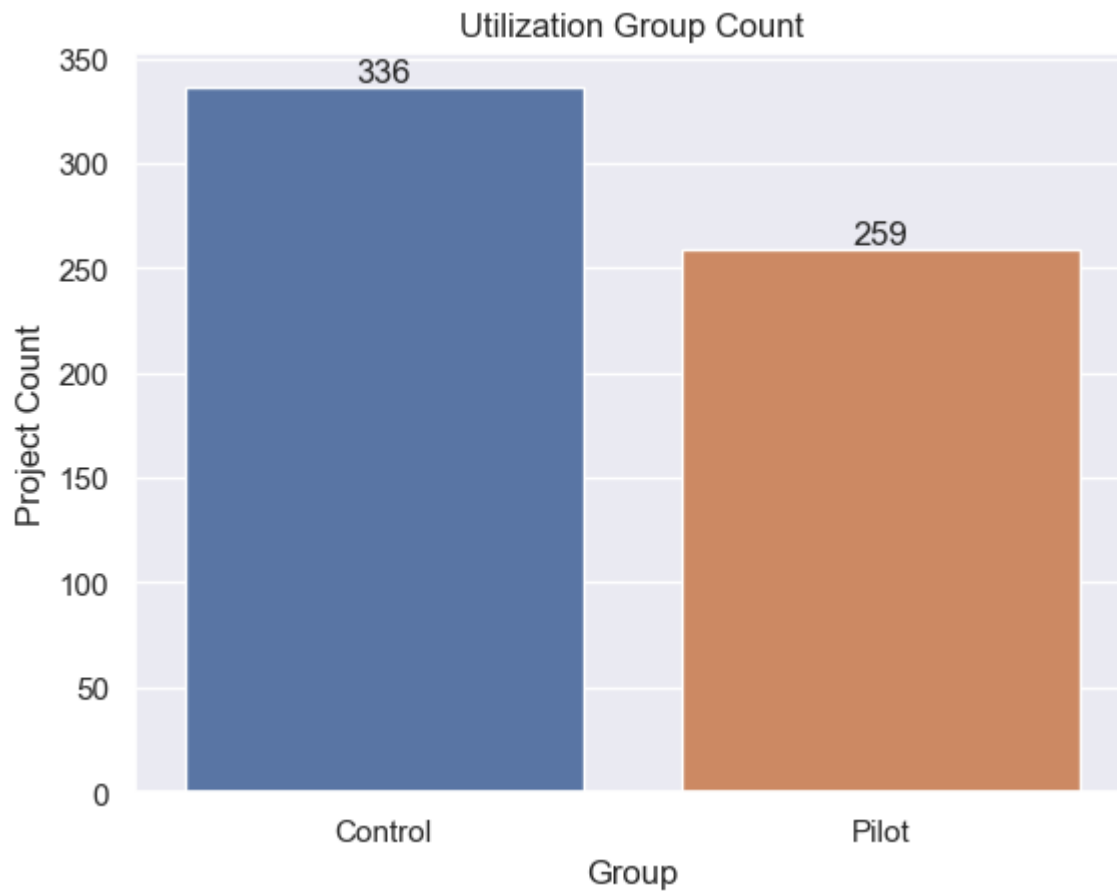
## Utilization Group Count Plot

```

In [6]: group_count = sns.countplot(data=utilization_df, x="isPilot")
group_count.set(xlabel='Group', ylabel = 'Project Count', title= 'Utilizatio

for i in group_count.containers:
    group_count.bar_label(i,)

```



## Control and Pilot Analysis

Overall HPP & DTV Quarter Statistics: Control Vs Pilot Groups

```
In [7]: #HPP StatsiticsOver Quarter
HPP_df = utilization_df[utilization_df['HPP'] > 0]
HPP_df = utilization_df.groupby(['isPilot', 'TransCSYearQuarter'])['HPP'].agg
HPP_df.columns = ['project_count', 'HPP_sum', 'HPP_mean', 'HPP_median']
HPP_df

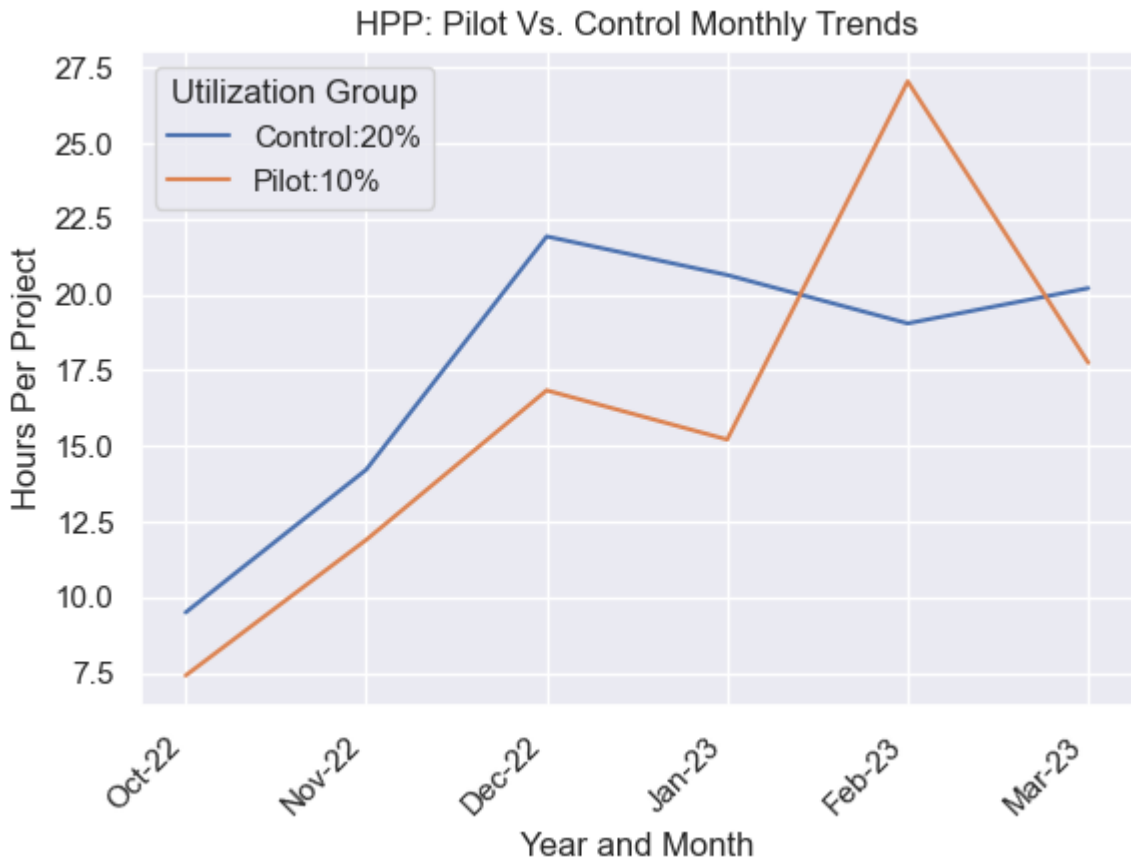
#DTV StatsiticsOver Quarter
DTV_df = utilization_df[utilization_df['DTV'] > 0]
DTV_df = utilization_df.groupby(['isPilot', 'TransCSYearQuarter'])['DTV'].agg
DTV_df.columns = ['DTV_sum', 'DTV_mean', 'DTV_median']
DTV_df

hpp_dtv_overall = pd.concat([HPP_df, DTV_df], axis=1).reindex(HPP_df.index)
hpp_dtv_overall
```

		project_count	HPP_sum	HPP_mean	HPP_median	DTV
isPilot	TransCSYearQuarter					
Control	2022-4	111	2011.56	18.122162	15.00	
	2023-1	225	4458.35	19.814889	15.77	2
Pilot	2022-4	95	1318.50	13.878947	10.00	
	2023-1	164	3481.22	21.226951	15.75	

HPP Lineplot

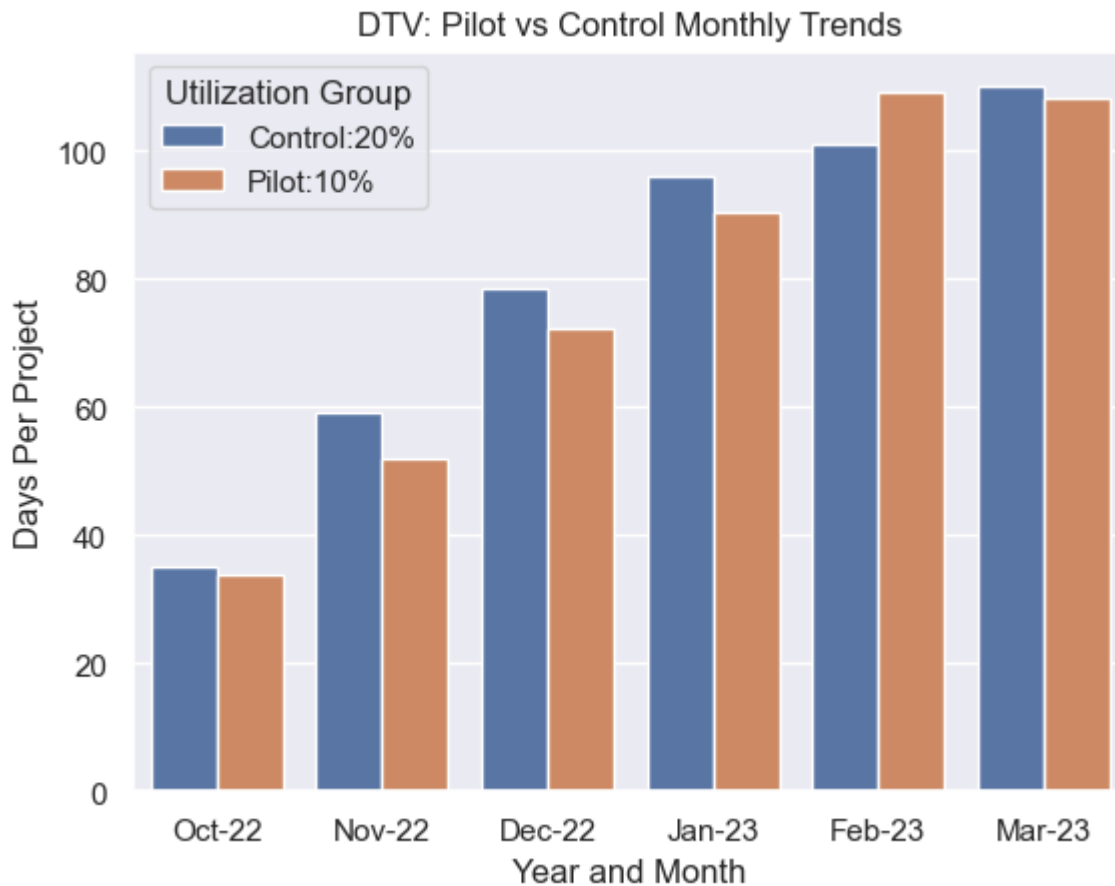
```
In [8]: #Plotted with aggregated mean of y values for each x wit 95% confidence inte
HPP_plot = sns.lineplot(data=utilization_df, x="TransCSYearMonth", y="HPP",
HPP_plot.legend(title='Utilization Group', labels=['Control:20%', 'Pilot:10%
HPP_plot.set(title= 'HPP: Pilot Vs. Control Monthly Trends', xlabel=' Year a
HPP_plot.figure.autofmt_xdate(rotation=45)
```



DTV Barplot

```
In [9]: #Plotted with aggregated mean of y values for each x wit 95% confidence inte
DTV_plot = sns.barplot(data=utilization_df, x="TransCSYearMonth", y="DTV", h
DTV_plot.set(title= 'DTV: Pilot vs Control Monthly Trends', xlabel=' Year an
DTV_plot.legend(title='Utilization Group', labels=['Control:20%', 'Pilot:10%
```

Out[9]: <matplotlib.legend.Legend at 0x28c00f4d0>



Coach Quarterly Productivity Statistics, derived from HPP

## Q1 2023: Activation Coach Productivity

```
In [10]: hpp_coach_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationCoach'])['Q1_project_count', 'Q1_HPP_median', 'Q1_HPP_sum']
hpp_coach_df_q1 = hpp_coach_df_q1.sort_values(by = ['isPilot', 'Q1_project_count'])
hpp_coach_df_q1

dtv_coach_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationCoach'])['Q1_DTV_median', 'Q1_DTV_sum', 'Q1_DTV_mean']
dtv_coach_df_q1 = dtv_coach_df_q1.sort_values(by = ['isPilot'], ascending=False)
dtv_coach_df_q1

coach_overall_q1 = pd.concat([hpp_coach_df_q1, dtv_coach_df_q1], axis=1).reindex(columns=coach_overall_q1.columns)
```

Out[10]:

		Q1_project_count	Q1_HPP_median	Q1_HPP_sum	Q1_HPP_mean
isPilot	ActivationCoach				
Pilot	Jenna Pingeon	9	0.00	9.00	1.00
	Denzel Alexander	8	26.75	208.00	26.00
	Josh Frandsen	8	24.75	156.00	19.50
	Austin Clary	7	9.00	80.00	11.43
	Diego Garcia	7	17.50	148.00	21.14
...	...	...	...	...	...
Control	Shweta Sahu	1	21.00	21.00	21.00
	Sola Kristie Broadrick	1	6.95	6.95	6.95
	Stephen Barlow	1	18.00	18.00	18.00
	Sushant Athawle	1	33.75	33.75	33.75
	Ted Pahl	1	55.75	55.75	55.75

151 rows x 7 columns

## Q4 2022: Activation Coach Productivity

```
In [11]: hpp_coach_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationCoach'])['Q4_project_count', 'Q4_HPP_median', 'Q4_HPP_sum', 'Q4_HPP_mean'].agg('mean').reset_index()
hpp_coach_df_q4 = hpp_coach_df_q4.sort_values(by = ['isPilot', 'Q4_project_count'], ascending=[True, False])
hpp_coach_df_q4

dtv_coach_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationCoach'])['Q4_DTV_median', 'Q4_DTV_sum', 'Q4_DTV_mean'].agg('mean').reset_index()
dtv_coach_df_q4 = dtv_coach_df_q4.sort_values(by = ['isPilot'], ascending=False)
dtv_coach_df_q4

coach_overall_q4 = pd.concat([hpp_coach_df_q4, dtv_coach_df_q4], axis=1).reset_index()
coach_overall_q4
```



Out[11]:

		Q4_project_count	Q4_HPP_median	Q4_HPP_sum	Q4_HPP
isPilot	ActivationCoach				
Pilot	Kristopher Horne	7	20.75	125.50	17.9
	Nicole Kolenic	5	17.50	83.00	16.6
	Anna Christofaro	4	8.75	50.50	12.6
	Austin Clary	4	11.00	46.00	11.5
	Samuel Burchill	4	9.00	51.50	12.8
...	...	...	...	...	
Control	Stephanie Castor	1	20.00	20.00	20.0
	Sue Medeiros	1	39.25	39.25	39.2
	Sujay Sanjeev	1	46.50	46.50	46.5
	Tina Harris	1	32.00	32.00	32.0
	Tracy Jahner	1	20.25	20.25	20.2

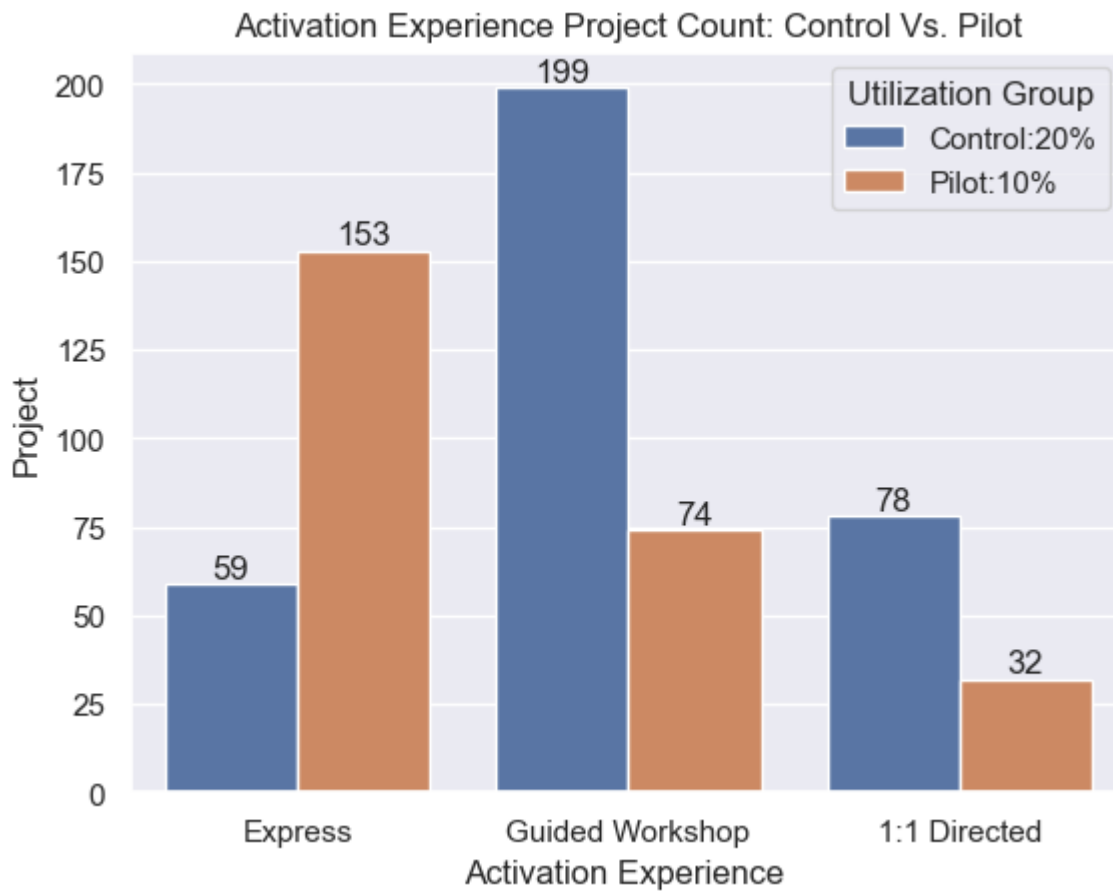
102 rows x 7 columns

## Part 2 Activation Experience: Pilot Vs. Control

### Activity Experience Distribution Check

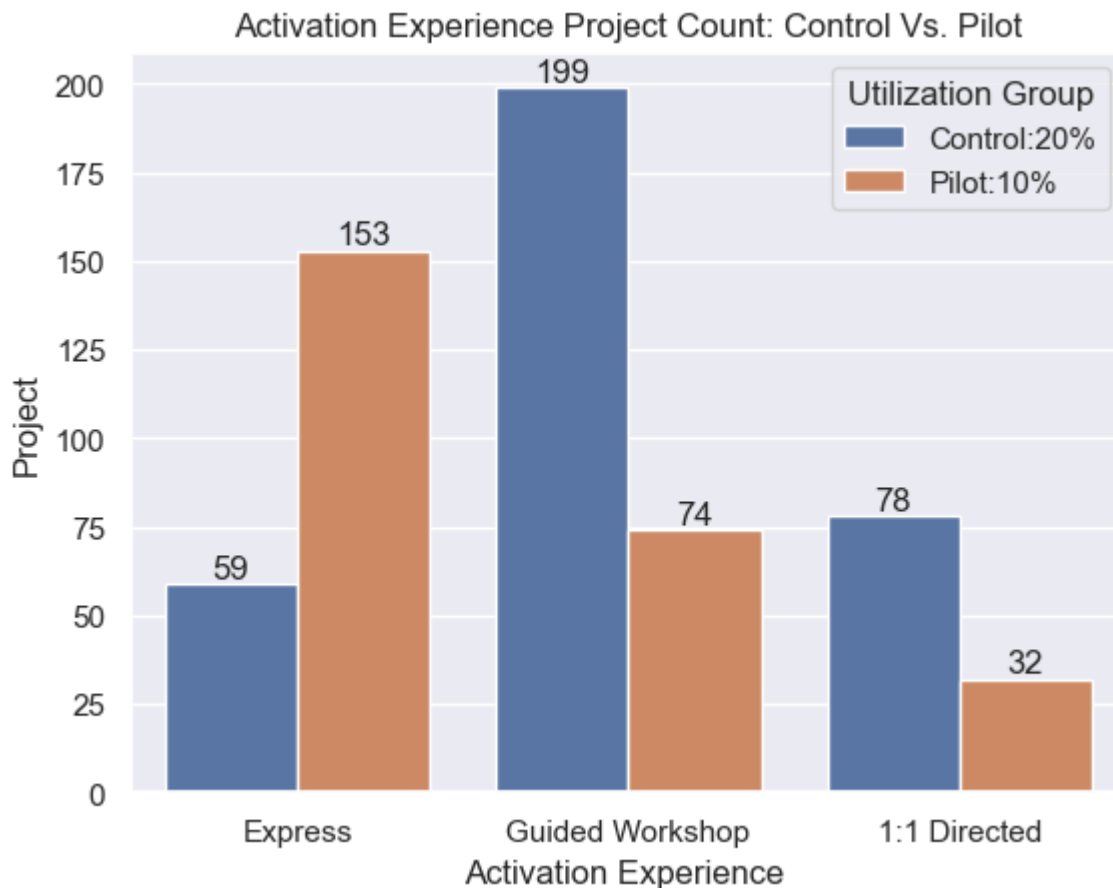
```
In [12]: experience_count = sns.countplot(data=utilization_df, x="ActivationExperience",
experience_count.legend(title='Utilization Group', labels=['Control:20%', 'Pilot:80%'])
experience_count.set(title= 'Activation Experience Project Count: Control Vs Pilot')

for i in experience_count.containers:
    experience_count.bar_label(i,)
```



```
In [13]: experience_count = sns.countplot(data=utilization_df, x="ActivationExperience")
experience_count.legend(title='Utilization Group', labels=['Control:20%', 'Pilot:10%'])
experience_count.set(title= 'Activation Experience Project Count: Control Vs Pilot')

for i in experience_count.containers:
    experience_count.bar_label(i,)
```



## Activation Experience: Overall Group Statistics

```
In [14]: #HPP StatsiticsOver Quarter
experience_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationExperience'])
experience_df_q1.columns = ['Q1_project_count', 'Q1_HPP_median', 'Q1_HPP_sum']
experience_df_q1 = experience_df_q1.sort_values(by = ['isPilot', 'ActivationExperience'])

#DTV StatsiticsOver Quarter
dtv_experience_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationExperience'])
dtv_experience_df_q1.columns = ['Q1_DTV_median', 'Q1_DTV_sum', 'Q1_DTV_mean']
dtv_experience_df_q1 = dtv_experience_df_q1.sort_values(by = ['isPilot', 'ActivationExperience'])

experience_hpp_dtv_overall = pd.concat([experience_df_q1, dtv_experience_df_q1])
experience_hpp_dtv_overall
```

		Q1_project_count	Q1_HPP_median	Q1_HPP_sum	Q1_HPP_mean
isPilot	ActivationExperience				
Pilot	Guided Workshop	56	22.500	1335.30	23.846
	Express	83	9.750	980.80	11.817
	1:1 Directed	25	34.250	1165.12	46.605
Control	Guided Workshop	138	13.575	2394.14	17.350
	Express	36	13.775	608.90	16.914
	1:1 Directed	51	24.000	1455.31	28.535

```

In [15]: #HPP StatsiticsOver Quarter
experience_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationExperience']).agg({'Q1_HPP_sum': 'sum', 'Q1_HPP_median': 'median', 'Q1_project_count': 'count'})
experience_df_q4.columns = ['Q4_project_count', 'Q4_HPP_median', 'Q4_HPP_sum']
experience_df_q4 = experience_df_q4.sort_values(by = ['isPilot', 'ActivationExperience'])

#DTV StatsiticsOver Quarter
dtv_experience_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationExperience']).agg({'Q4_DTV_sum': 'sum', 'Q4_DTV_mean': 'mean', 'Q4_DTV_median': 'median'})
dtv_experience_df_q4.columns = ['Q4_DTV_sum', 'Q4_DTV_mean', 'Q4_DTV_median']
dtv_experience_df_q4 = dtv_experience_df_q4.sort_values(by = ['isPilot', 'ActivationExperience'])

experience_hpp_dtv_overall4 = pd.concat([experience_df_q4, dtv_experience_df_q4], axis=1)
experience_hpp_dtv_overall4

```

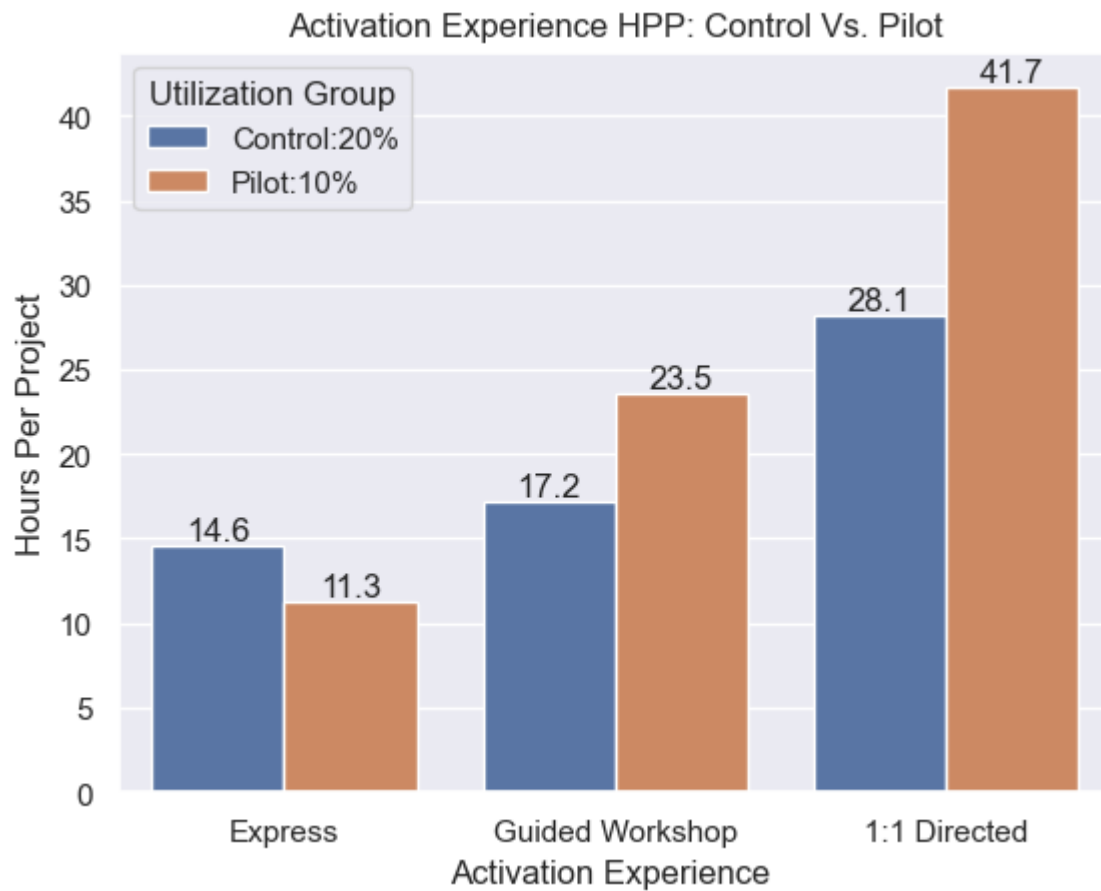
		Q4_project_count	Q4_HPP_median	Q4_HPP_sum	Q4_HPP_mean
isPilot	ActivationExperience				
Pilot	Guided Workshop	18	22.00	406.00	22.556
	Express	70	8.25	744.25	10.632
	1:1 Directed	7	22.25	168.25	24.036
Control	Guided Workshop	61	15.00	1019.41	16.712
	Express	23	11.00	252.65	10.985
	1:1 Directed	27	25.00	739.50	27.389

## Activation Experience: HPP

```

In [16]: HPP_experience_plot = sns.barplot(data=utilization_df, x="ActivationExperience", y="Q4_HPP_mean", hue="isPilot")
HPP_experience_plot.legend(title='Utilization Group', labels=['Control:20%', 'Pilot:80%'])
HPP_experience_plot.set(title= 'Activation Experience HPP: Control Vs. Pilot')
for i in HPP_experience_plot.containers:
    HPP_experience_plot.bar_label(i, fmt= '%.1f')

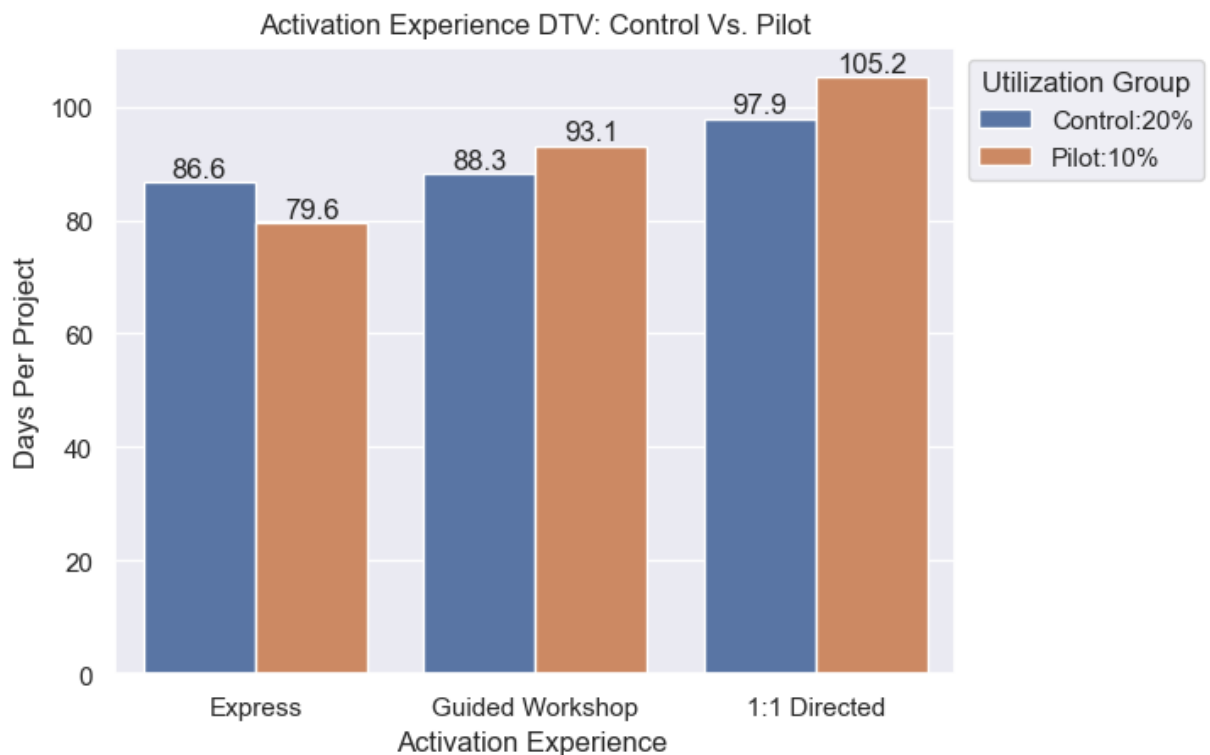
```



## Activation Experience: DTV

```
In [17]: DTV_experience_plot = sns.barplot(data=utilization_df, x="ActivationExperien
DTV_experience_plot.legend(title='Utilization Group', labels=['Control:20%',
DTV_experience_plot.set(title= 'Activation Experience DTV: Control Vs. Pilot

for i in DTV_experience_plot.containers:
    DTV_experience_plot.bar_label(i,fmt= '%.1f')
```



## Activation Experience: Activation Coach Productivity

### Q4 2022 Tables

```
In [18]: express_df_q4 = utilization_df_q4[utilization_df_q4['ActivationExperience']!=
express_df_q4 = express_df_q4.groupby(['isPilot', 'ActivationCoach'])['HPP']
express_df_q4.columns = ['express_project_count', 'express_HPP_median', 'express_DTV_median', 'express_DTV_sum']
express_df_q4 = express_df_q4.sort_values(by = ['isPilot', 'express_project_count'], ascending=[True, False])

dtv_express_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationCoach'])['HPP']
dtv_express_df_q4.columns = ['express_DTV_median', 'express_DTV_sum', 'express_HPP_median', 'express_HPP_sum']
dtv_express_df_q4 = dtv_express_df_q4.sort_values(by = ['isPilot'], ascending=[True])

express_overall_q4 = pd.concat([express_df_q4, dtv_express_df_q4], axis=1).reset_index()
express_overall_q4
```

Out[18]:

		express_project_count	express_HPP_median	express_HPP_
isPilot	ActivationCoach			
Pilot	Kristopher Horne	4.0	10.000	5
	Nicole Kolenic	4.0	17.500	6
	Austin Clary	3.0	11.000	3
	Charlene Bystrom	3.0	16.250	4
	Jaffar Mohammed Siddiqui	3.0	7.500	2
	Michele Testi	3.0	11.500	3
	Samuel Burchill	3.0	9.000	2
	Abi Sweet	2.0	6.000	1
	Akilesh R	2.0	11.625	2
	Angel Bradshaw	2.0	12.750	2
	Annie Leidner	2.0	4.625	
	Britney Peoples	2.0	12.375	2
	Diego Garcia	2.0	20.500	4
	Gerick Kean Lumapak	2.0	10.750	2
	Josh Frandsen	2.0	4.000	
	Lexis Julsrud-Wertjes	2.0	5.875	
	Marcky Azarcon	2.0	42.750	8
	Nathaniel Rondina Alivio	2.0	12.500	2
	Prisilla Paul	2.0	7.000	1
	Yuri Alfante	2.0	6.875	1
	Alex Mavros	1.0	10.000	1
	Alexis Blue	1.0	4.750	
	Alyssa Gray	1.0	1.750	
	Anita Verma	1.0	3.500	
	Anna Christofaro	1.0	5.500	
	Cassie Willetts	1.0	31.000	3
	Christina Chalepoudi	1.0	4.000	

		express_project_count	express_HPP_median	express_HPP_
isPilot	ActivationCoach			
	Daniel Kihm	1.0	3.500	
	Danielle Hegge	1.0	5.000	
	Ian Pajarillo	1.0	12.500	1
	Joanna Jerica Flores	1.0	4.500	
	Jonathan Catungal	1.0	2.000	
	Julie Bruner	1.0	3.250	
	Kacie Johnson	1.0	4.500	
	Kyle Comer	1.0	21.000	2
	Lance Williams	1.0	7.000	
	PJ Schoeny	1.0	0.000	
	Peter Griganavicius	1.0	6.500	
	Rizwana Bellary Kolmi	1.0	9.000	
	Sasidhar Swarna	1.0	5.500	
	Tracy Jahner	1.0	7.000	
Control	Diego Garcia	4.0	15.250	5
	Anita Verma	2.0	10.450	2
	Joanna Jerica Flores	2.0	16.625	3
	Rizwana Bellary Kolmi	2.0	12.000	2
	Angel Bradshaw	1.0	6.250	
	Ben Boddy	1.0	15.000	1
	Danielle Hegge	1.0	4.000	
	Ferdie Malapit	1.0	7.500	
	Jaffar Mohammed Siddiqui	1.0	8.000	
	Jessica Pearson	1.0	2.750	
	Jonathan Catungal	1.0	4.500	
	Jose Remeterio Jr.	1.0	2.000	



		express_project_count	express_HPP_median	express_HPP_
isPilot	ActivationCoach			
	Lexis Julsrud-Wertjes	1.0	4.500	
	Maggi Manoukian	1.0	35.500	3
	Ouafaa Chaouki	1.0	13.000	1
	Paul Walker	1.0	3.500	
	Prisilla Paul	1.0	11.000	

```
In [19]: guided_df_q4 = utilization_df_q4[utilization_df_q4['ActivationExperience']=='
guided_df_q4 = guided_df_q4.groupby(['isPilot', 'ActivationCoach'])['HPP'].a
guided_df_q4.columns = ['guided_project_count', 'guided_HPP_median', 'guided_
guided_df_q4 = guided_df_q4.sort_values(by = ['isPilot', 'guided_project_cou

dtv_guided_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationCoach'])
dtv_guided_df_q4.columns = ['guided_DTV_median', 'guided_DTV_sum', 'guided_D
dtv_guided_df_q4 = dtv_guided_df_q4.sort_values(by = ['isPilot'], ascending=

guided_overall_df_q4 = pd.concat([guided_df_q4, dtv_guided_df_q4], axis=1).r
guided_overall_df_q4
```

Out[19]:

		guided_project_count	guided_HPP_median	guided_HPP_sui
isPilot	ActivationCoach			
Pilot	Kristopher Horne	3.0	22.000	72.7
	Christine Bonvini	2.0	17.250	34.5
	Denzel Alexander	2.0	14.500	29.0
	Marlo Korbel	2.0	46.125	92.2
	Monica Miller	2.0	22.000	44.0
	Anna Christofaro	1.0	9.000	9.0
	Austin Clary	1.0	11.000	11.0
	Cassie Willetts	1.0	12.000	12.0
	Daniel Kihm	1.0	25.500	25.5
	Nicole Kolenic	1.0	17.500	17.5
	Samuel Burchill	1.0	28.500	28.5
	Valerie Tema	1.0	30.000	30.0
Control	Jaffar Mohammed Siddiqui	4.0	23.000	88.0
	Alexis Blue	3.0	9.750	32.0
	Annie Leidner	3.0	8.000	35.0
	Rizwana Bellary Kolmi	3.0	13.000	40.0
	Rose Platon	3.0	6.000	17.7
	Shankaran Pantula	3.0	29.250	83.2
	Alisa DiGeronimo	2.0	17.500	35.0
	Anita Verma	2.0	16.100	32.2
	Glenda Servidad	2.0	40.375	80.7
	Jaime Fulgosino	2.0	6.000	12.0
	Jessica Pearson	2.0	11.625	23.2
	Jose Remeterio Jr.	2.0	12.750	25.5
	Kyle Comer	2.0	7.750	15.5

		guided_project_count	guided_HPP_median	guided_HPP_sui
isPilot	ActivationCoach			
	Leslia Fernandes Barnes	2.0	14.625	29.2
	Lexis Julsrud- Wertjes	2.0	13.000	26.0
	Michele Testi	2.0	8.500	17.0
	Akilesh R	1.0	27.250	27.2
	Ayra Cornel	1.0	16.000	16.0
	Barbara Davis	1.0	19.500	19.5
	Brett Johnson	1.0	16.000	16.0
	Caren Lea Davis	1.0	10.010	10.0
	Christian Schlaefer	1.0	22.000	22.0
	Danielle Hegge	1.0	11.750	11.7
	James Tettmar	1.0	4.500	4.5
	Jessica Colley	1.0	14.000	14.0
	John DePuy	1.0	0.000	0.0
	Jonathon Waters	1.0	32.500	32.5
	Lizzie Bolton	1.0	11.000	11.0
	Paul Walker	1.0	29.000	29.0
	Prisilla Paul	1.0	16.000	16.0
	Sarah Dastas	1.0	32.000	32.0
	Seerat Dhawan	1.0	10.250	10.2
	Seth Bresky	1.0	15.750	15.7
	Shannon Santopietro	1.0	20.750	20.7
	Stephanie Castor	1.0	20.000	20.0
	Sujay Sanjeev	1.0	46.500	46.5
	Tina Harris	1.0	32.000	32.0
	Tracy Jahner	1.0	20.250	20.2

```
In [20]: oneonone_df_q4 = utilization_df_q4[utilization_df_q4['ActivationExperience']]
oneonone_df_q4 = oneonone_df_q4.groupby(['isPilot', 'ActivationCoach'])['HPP
oneonone_df_q4.columns = ['oneonone_project_count', 'oneonone_HPP_median', 'c
```

```

oneonone_df_q4 = oneonone_df_q4.sort_values(by = ['isPilot', 'oneonone_project_count'])
oneonone_df_q4

dtv_oneonone_df_q4 = utilization_df_q4.groupby(['isPilot', 'ActivationCoach'])
dtv_oneonone_df_q4.columns = ['oneonone_DTV_median', 'oneonone_DTV_sum', 'oneonone_DTV_count']
dtv_oneonone_df_q4 = dtv_oneonone_df_q4.sort_values(by = ['isPilot'], ascending=False)
dtv_oneonone_df_q4

oneonone_overall_df_q4 = pd.concat([oneonone_df_q4, dtv_oneonone_df_q4], axis=1)
oneonone_overall_df_q4

```

Out[20]:

		oneonone_project_count	oneonone_HPP_median	oneonone_DTV_count
isPilot	ActivationCoach			
Pilot	Adam Haas	3.0	22.250	
	Anna Christofaro	2.0	18.000	
	Josh Frandsen	1.0	11.000	
	Valerie Tema	1.0	40.750	
Control	Kokila Raj	3.0	49.000	
	Alexis Blue	2.0	6.500	
	Annie Leidner	2.0	8.875	
	Brett Johnson	2.0	26.000	
	Lynae Peterson	2.0	41.125	
	Ouafaa Chaouki	2.0	49.250	
	PJ Schoeny	2.0	1.750	
	Prisilla Paul	2.0	16.125	
	Amiel Dela Rosa	1.0	16.500	
	Anita Verma	1.0	11.000	
	Barbara Davis	1.0	27.250	
	Elwel Gardaya	1.0	49.500	
	Harika Voota	1.0	43.500	
	Jes Gilman	1.0	16.500	
	Ken Sterner	1.0	36.250	
	Maggi Manoukian	1.0	25.000	
	Michele Testi	1.0	27.000	
	Sue Medeiros	1.0	39.250	

## Q1 2023 Tables

```
In [21]: express_df_q1 = utilization_df_q1[utilization_df_q1['ActivationExperience']=='express']
express_df_q1 = express_df_q1.groupby(['isPilot', 'ActivationCoach'])['HPP'].agg('mean')
express_df_q1.columns = ['express_project_count', 'express_HPP_median', 'express_HPP_sum']
express_df_q1 = express_df_q1.sort_values(by = ['isPilot', 'express_project_count'], ascending=[True, False])

dtv_express_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationCoach'])['DTV'].agg('mean')
dtv_express_df_q1.columns = ['express_DTV_median', 'express_DTV_sum', 'express_DTV_count']
dtv_express_df_q1 = dtv_express_df_q1.sort_values(by = ['isPilot'], ascending=[True])

express_overall_q1 = pd.concat([express_df_q1, dtv_express_df_q1], axis=1).reset_index()
```

```
Out[21]:
```

		express_project_count	express_HPP_median	express_HPP_sum
isPilot	ActivationCoach			
Pilot	Jenna Pingeon	9.0	0.00	
	Diego Garcia	7.0	17.50	14
	Nathaniel Rondina Alivio	5.0	17.50	10
	Britney Peoples	4.0	12.25	5
	Kristopher Horne	4.0	16.00	7
...	...	...	...	
Control	Marcky Azarcon	1.0	33.00	3
	Ruvie Dianne Cudo	1.0	9.00	
	Sarah Dastas	1.0	26.50	2
	Sunitha Vadivelu	1.0	33.50	3
	Tracy Jahner	1.0	11.00	1

64 rows x 7 columns

```
In [22]: guided_df_q1 = utilization_df_q1[utilization_df_q1['ActivationExperience']=='guided']
guided_df_q1 = guided_df_q1.groupby(['isPilot', 'ActivationCoach'])['HPP'].agg('mean')
guided_df_q1.columns = ['guided_project_count', 'guided_HPP_median', 'guided_HPP_sum']
guided_df_q1 = guided_df_q1.sort_values(by = ['isPilot', 'guided_project_count'], ascending=[True, False])

dtv_guided_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationCoach'])['DTV'].agg('mean')
dtv_guided_df_q1.columns = ['guided_DTV_median', 'guided_DTV_sum', 'guided_DTV_count']
dtv_guided_df_q1 = dtv_guided_df_q1.sort_values(by = ['isPilot'], ascending=[True])

guided_overall_df_q1 = pd.concat([guided_df_q1, dtv_guided_df_q1], axis=1).reset_index()
```

Out [22]:

		guided_project_count	guided_HPP_median	guided_HPP_sui
isPilot	ActivationCoach			
Pilot	Adam Haas	5.0	43.75	217.7
	Denzel Alexander	5.0	17.75	102.2
	Austin Clary	4.0	7.00	38.0
	Monica Miller	4.0	20.25	89.0
	Nicole Kolenic	4.0	13.75	59.2
...	...	...	...	
Control	Shannon Santopietro	1.0	30.50	30.5
	Shweta Sahu	1.0	21.00	21.0
	Stephen Barlow	1.0	18.00	18.0
	Tariq Mohammed Mehkri	1.0	15.00	15.0
	Ted Pahl	1.0	55.75	55.7

83 rows × 7 columns

```
In [23]: oneonone_df_q1 = utilization_df_q1[utilization_df_q1['ActivationExperience']
oneonone_df_q1 = oneonone_df_q1.groupby(['isPilot', 'ActivationCoach'])['HPP
oneonone_df_q1.columns = ['oneonone_project_count', 'oneonone_HPP_median', 'c
oneonone_df_q1 = oneonone_df_q1.sort_values(by = ['isPilot', 'oneonone_proje
oneonone_df_q1

dtv_oneonone_df_q1 = utilization_df_q1.groupby(['isPilot', 'ActivationCoach']
dtv_oneonone_df_q1.columns = ['oneonone_DTV_median', 'oneonone_DTV_sum', 'on
dtv_oneonone_df_q1 = dtv_oneonone_df_q1.sort_values(by = ['isPilot'], ascend
dtv_oneonone_df_q1

oneonone_overall_df_q1 = pd.concat([oneonone_df_q1, dtv_oneonone_df_q1], axi
oneonone_overall_df_q1
```

Out [23]:

		oneonone_project_count	oneonone_HPP_median	oneonone_
isPilot	ActivationCoach			
Pilot	Christopher Smith	4.0	46.125	
	Denzel Alexander	3.0	34.250	
	Brandy Lane	2.0	88.920	
	Christopher Doyle	2.0	19.240	
	Daniel Forester	2.0	21.500	
	Jacquelyn Wager	2.0	25.300	
	Josh Frandsen	2.0	29.250	
	Adam Haas	1.0	31.250	
	Alice Whalen	1.0	0.000	
	Anna Christofaro	1.0	37.500	
	Austin Clary	1.0	12.000	
	Julie Bruner	1.0	79.250	
	Odalys Marinas	1.0	157.000	
	Richard Needham	1.0	34.500	
	Samuel Burchill	1.0	35.000	
Control	Christian Schlaefer	3.0	8.750	
	Cindy Schafer	3.0	18.500	
	Nicole Pulido	3.0	35.500	
	Paul Walker	3.0	18.000	
	Brett Johnson	2.0	36.000	
	Diana Chris Mantua	2.0	31.500	
	Mary Fahey	2.0	16.500	
	Michael Rezler	2.0	31.125	
	Shannon Santopietro	2.0	24.625	
	Tina Harris	2.0	49.500	
	Alexander Agbaglud	1.0	40.010	

	oneonone_project_count	oneonone_HPP_median	oneonone_
isPilot	ActivationCoach		
	Angela Armstrong	1.0	31.750
	Ayra Cornel	1.0	19.000
	Caren Lea Davis	1.0	40.000
	Danielle Hegge	1.0	5.750
	Elise Schroeffer	1.0	44.250
	Elwel Gardaya	1.0	29.750
	Harika Voota	1.0	18.500
	Joanna Jerica Flores	1.0	15.500
	Justin Smith	1.0	18.000
	Justin Steabner	1.0	41.550
	Kayla Shreve	1.0	41.250
	Ken Sterner	1.0	24.000
	Kevin Lamb	1.0	14.750
	Leslia Fernandes Barnes	1.0	47.000
	Lisa Lawson	1.0	76.800
	Mi Richie Forio	1.0	21.750
	Michele Testi	1.0	17.000
	Ouafaa Chaouki	1.0	58.500
	PJ Schoeny	1.0	5.000
	Rajiv Raj	1.0	52.000
	Rizwana Bellary Kolmi	1.0	18.250
	Rynal Stanley	1.0	7.500
	Sara Beth Vilbig	1.0	17.750
	Sola Kristie Broadrick	1.0	6.950
	Sunitha Vadivelu	1.0	26.500
	Sushant Athawle	1.0	33.750