Project - Group 11 (1)

August 12, 2021

[44]:
```
%config InlineBackend.figure_format = 'retina'
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib as mpl
%matplotlib inline
mpl.rcParams['figure.dpi'] = 600
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('png', 'pdf')
```

[29]:
```
df=pd.read_csv('HR_Employee_Data.csv')
```

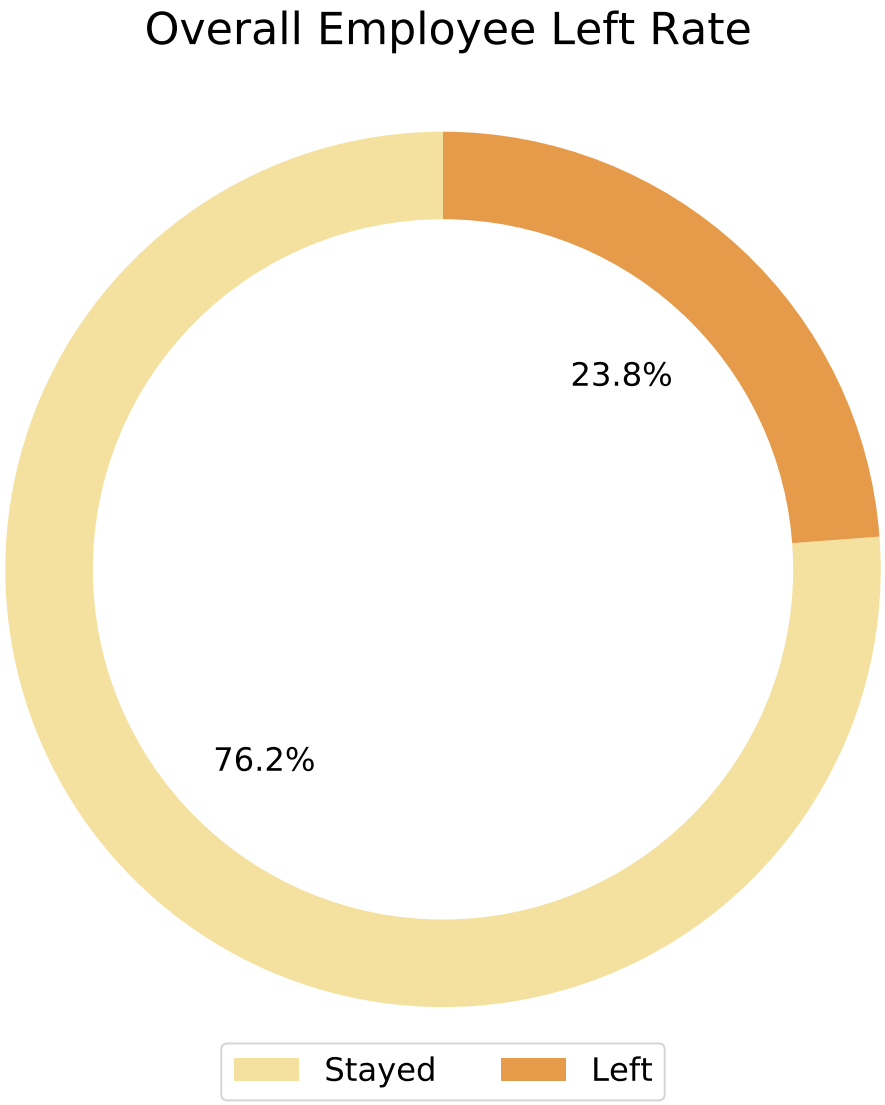
[30]:
```
df.head()
```

[30]:

	Emp_Id	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	Department	salary
0	IND02438	38%	53%	2	157	3	0	1	0	sales	low
1	IND28133	80%	86%	5	262	6	0	1	0	sales	medium
2	IND07164	11%	88%	7	272	4	0	1	0	sales	medium
3	IND30478	72%	87%	5	223	5	0	1	0	sales	low
4	IND24003	37%	52%	2	159	3	0	1	0	sales	low

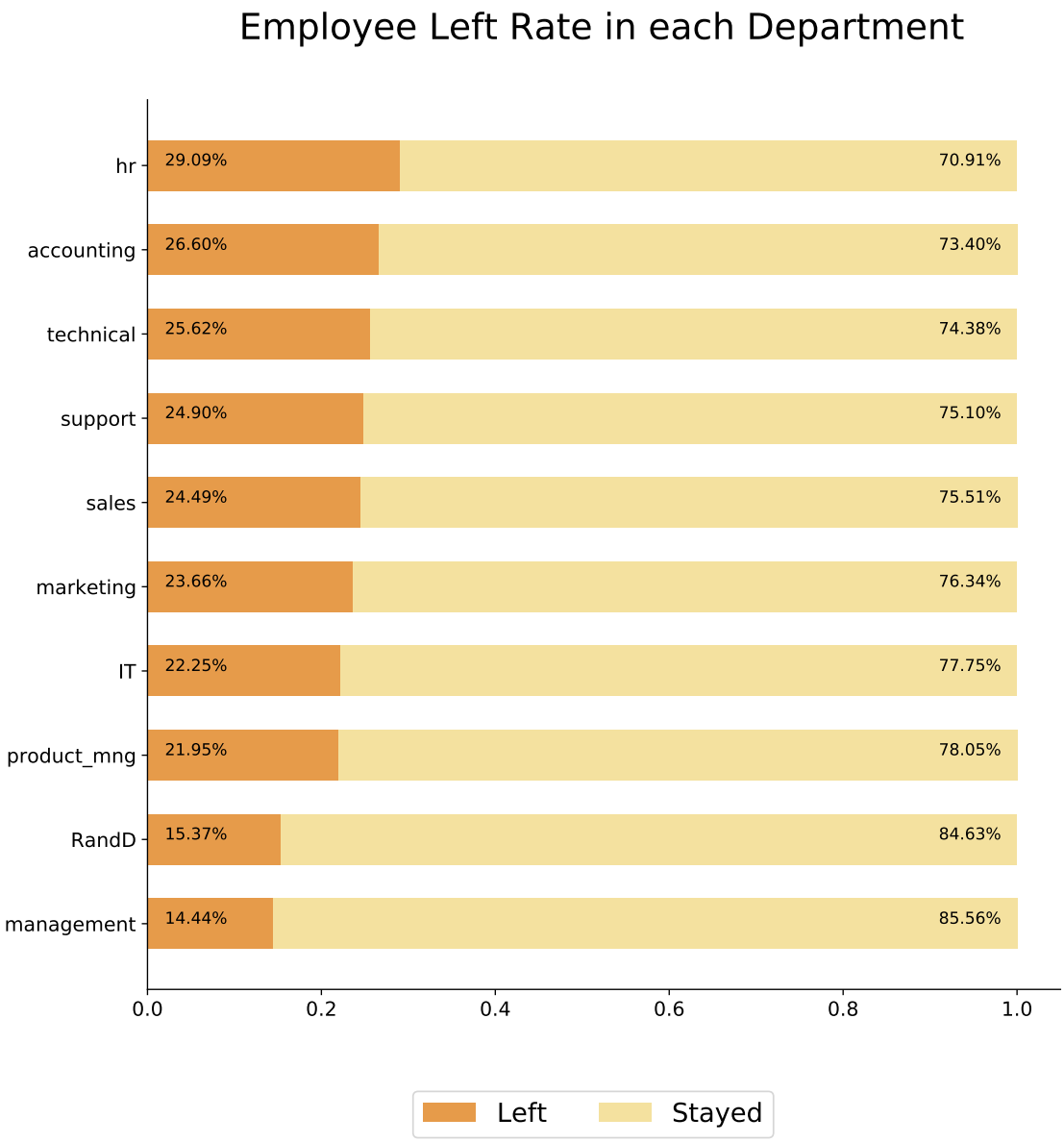
0.1 Data Overivew

[45]:
```
left=df['left'].value_counts()
leftsum=left[0]+left[1]
left_y=[left[0]/leftsum*100,left[1]/leftsum*100]
labels = ['', '']


fig,ax=plt.subplots(figsize=(10,10))

ax.pie(left_y, labels=labels, autopct='%1.
 ↪1f%%',colors=['#f4e19f','#e69b4a'],startangle=90,textprops={'fontsize': 16})
centre_circle = plt.Circle((0,0),0.80,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

ax.text(-1.1, 1.2, "             Overall Employee Left␣
 ↪Rate",fontdict={'fontsize': 22})

ax.legend(loc=8, labels=['Stayed', 'Left'],ncol=2,prop={'size': 16})

plt.show()

# fig.savefig('./Overall Employee Left Rate.png')
```



[46]:
```
departmentnumber=df.groupby('Department').count()['left']
department=df.groupby('Department').sum()['left'].values
departmentname=df.groupby('Department').count()['left'].index


left=department/departmentnumber
stayed=1-left

datasource=zip(departmentname,left,stayed)
newdf=pd.DataFrame(datasource,columns=['Dept','Left','Stayed']).
 ↪sort_values('Left',ascending=True)
```

```
#we need segment here (stayed,left) for each department

x = newdf['Left'].to_list()
y = newdf['Stayed'].to_list()

fig,ax=plt.subplots(figsize=(10,10))
ax.barh(range(len(x)),x,label='Left',color='#e69b4a',height = 0.6)
ax.barh(range(len(x)),y,left=x,label='stayed',color='#f4e19f',height = 0.6)

ax.set_yticks(range(len(x)))
ax.set_yticklabels(newdf['Dept'])

ax.tick_params(axis='both', which='major', labelsize=12)

ax.text(0, 10.5, '        Employee Left Rate in each␣
 ↪Department',fontdict={'fontsize': 22})

for i, v in enumerate(x):
    ax.text(0+0.02, i , '{:.2f}%'.format(100*v), color='black')

for i, v in enumerate(y):
    ax.text(1-0.09, i, '{:.2f}%'.format(100*v), color='black')

ax.legend(bbox_to_anchor=(0.7, -0.1), labels=['Left',␣
 ↪'Stayed'],ncol=2,prop={'size': 16})
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)


# fig.savefig('./Employee Left Rate in each Department.png')
plt.show()
```



1 Salary

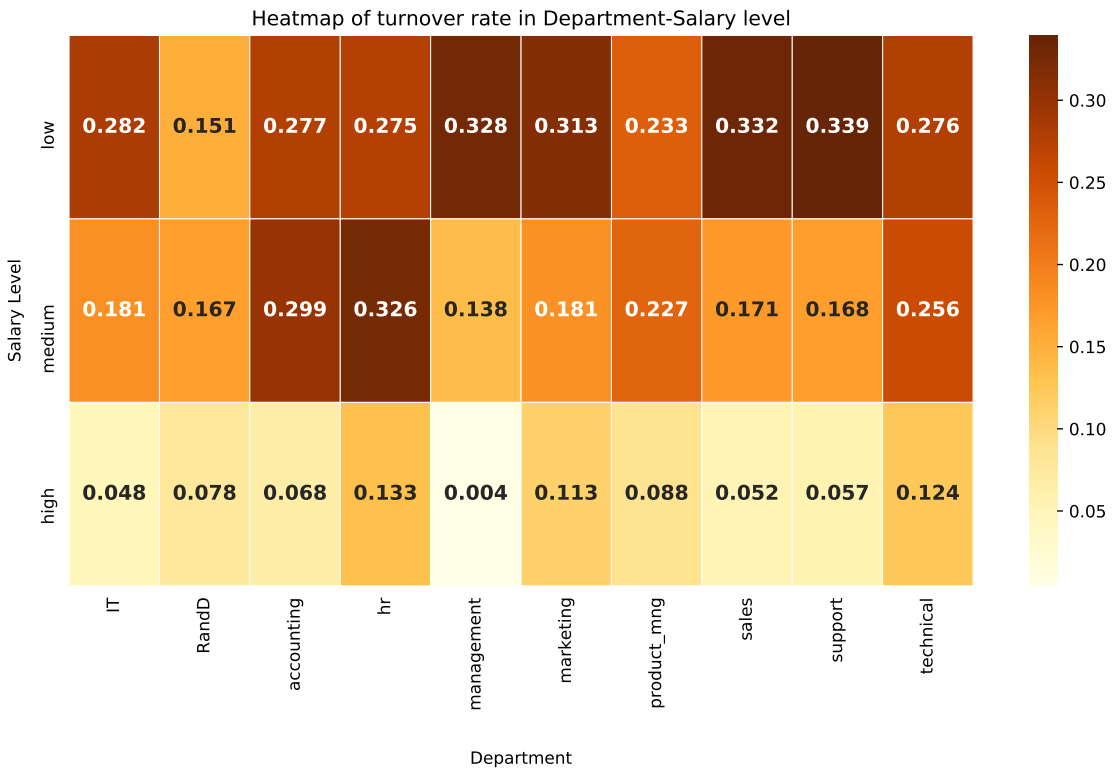
explain

Abnormal Group: R&D, Accounting, HR Normal Group: The other 7 departments

Where is abnormal? Turnover rate is **higher** in **medium** salary level than in **low** salary level.

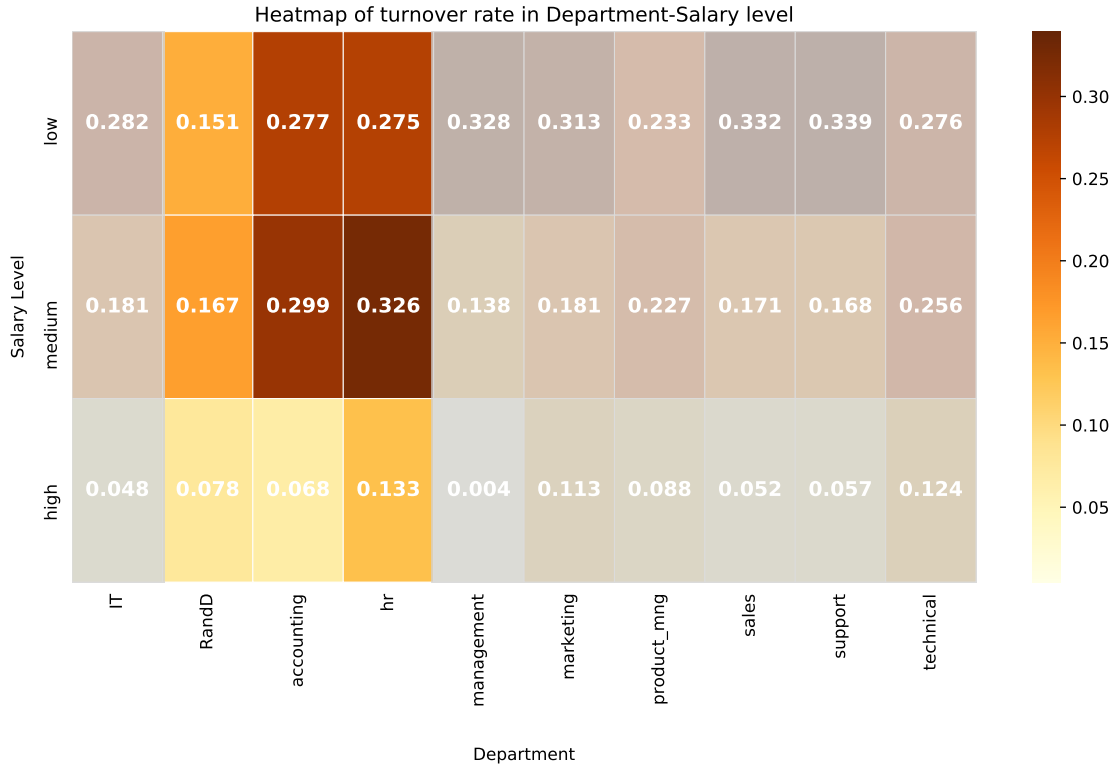
[47]:
```
tab3=df.groupby(['salary','Department'])['left'].sum()/df.
↪groupby(['salary','Department'])['left'].count()
tab3=tab3.reset_index()
```

```
tab3['salary'] =tab3['salary'].astype('category')
tab3['salary'] = tab3['salary'].cat.reorder_categories(['low','medium','high'])
tab4=tab3.pivot("salary", "Department")
fig, ax = plt.subplots(figsize=(12,6))
ax = sns.heatmap(tab4,cmap="YlOrBr",annot =True,fmt=".3f",xticklabels=tab3.
 ↪Department.unique(),linewidths=.5
                 ,annot_kws={'size':12,'weight':'bold'})
ax.set_xlabel('Department')
ax.set_ylabel('Salary Level')
ax.set_title('Heatmap of turnover rate in Department-Salary level')
#ax.add_patch(plt.Rectangle((0, 0), 1, 3, fill=True,color='lightgrey',alpha=0.
 ↪8, edgecolor='white', lw=3))
#ax.add_patch(plt.Rectangle((4, 0), 6, 3, fill=True,color='lightgrey',alpha=0.
 ↪8, edgecolor='white', lw=3))
ax.yaxis.set_label_coords(-0.05,0.5)
ax.xaxis.set_label_coords(0.5,-0.3)
ax.tick_params(axis='x', bottom=False)
ax.tick_params(axis='y', left=False)

# fig.savefig('./Heatmap of turnover rate in Department-Salary level.png')

plt.show()
```



[48]:
```
tab3['salary'] =tab3['salary'].astype('category')
tab3['salary'] = tab3['salary'].cat.reorder_categories(['low','medium','high'])
tab4=tab3.pivot("salary", "Department")
fig, ax = plt.subplots(figsize=(12,6))
ax = sns.heatmap(tab4,cmap="YlOrBr",annot =True,fmt=".3f",xticklabels=tab3.
 ↪Department.unique(),linewidths=.5
                 ,annot_kws={'size':12,'color':'white','weight':'bold'})
ax.set_xlabel('Department')
ax.set_ylabel('Salary Level')
ax.set_title('Heatmap of turnover rate in Department-Salary level')
ax.add_patch(plt.Rectangle((0, 0), 1, 3, fill=True,color='lightgrey',alpha=0.8,␣
 ↪edgecolor='white', lw=3))
ax.add_patch(plt.Rectangle((4, 0), 6, 3, fill=True,color='lightgrey',alpha=0.8,␣
 ↪edgecolor='white', lw=3))
ax.yaxis.set_label_coords(-0.05,0.5)
ax.xaxis.set_label_coords(0.5,-0.3)
ax.tick_params(axis='x', bottom=False)
ax.tick_params(axis='y', left=False)
# fig.savefig('./Heatmap of turnover rate in Department-Salary level2.png')
plt.show()
```

```
<ipython-input-48-35dac1b19417>:10: UserWarning: Setting the 'color' property
will override the edgecolor or facecolor properties.
  ax.add_patch(plt.Rectangle((0, 0), 1, 3,
fill=True,color='lightgrey',alpha=0.8, edgecolor='white', lw=3))
<ipython-input-48-35dac1b19417>:11: UserWarning: Setting the 'color' property
will override the edgecolor or facecolor properties.
  ax.add_patch(plt.Rectangle((4, 0), 6, 3,
fill=True,color='lightgrey',alpha=0.8, edgecolor='white', lw=3))
```



2 Grouping

define the grouping rule

3 Satisfaction Level

Speicial High paying group:: High paying staff who left the company have a dramatically lower satisfaction rate of 0.306 compared with an extremely high satisfaction rate of those who stay in the company 0.652, it's 53.8% drop. While for normal_group the satisfaction rate of the left people only drops 30%

Solution: For the Special group, we need to pay more attention to the high-paying group and improve their feedback based on their characteristic.

[35]:
```
df4
```

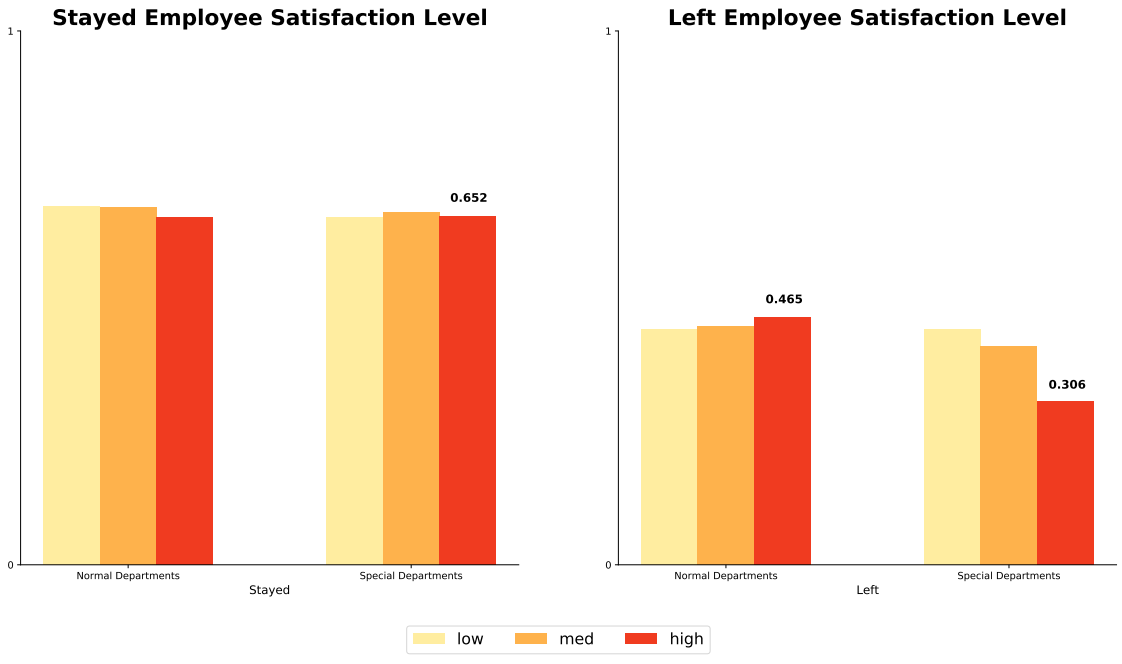
[35]:
```
                          satisfaction_level  last_evaluation  \
left salary dep
0    high   normal_group             0.651860         0.704890
            special_group            0.652000         0.732581
     low    normal_group             0.671320         0.716289
            special_group            0.650912         0.720974
```

```
     medium normal_group            0.670304         0.716513
            special_group           0.660991         0.709657
1    high   normal_group            0.463284         0.649851
            special_group           0.306000         0.618000
     low    normal_group            0.441184         0.719637
            special_group           0.441748         0.696301
     medium normal_group            0.446195         0.729104
            special_group           0.410108         0.707706

                           number_project  average_montly_hours  \
left salary dep
0    high   normal_group         3.770000            199.435000
            special_group        3.916129            206.238710
     low    normal_group         3.785830            198.506577
            special_group        3.778052            199.209618
     medium normal_group         3.793183            199.523031
            special_group        3.759848            197.510801
1    high   normal_group         3.492537            193.567164
            special_group        3.266667            191.000000
     low    normal_group         3.848910            206.973520
            special_group        3.735772            203.951220
     medium normal_group         3.933526            210.896917
            special_group        3.835125            204.824373

                           time_spend_company  Work_accident  \
left salary dep
0    high   normal_group             3.767000       0.164000
            special_group            3.219355       0.180645
     low    normal_group             3.259635       0.184399
            special_group            3.223181       0.180025
     medium normal_group             3.447490       0.171350
            special_group            3.372300       0.151207
1    high   normal_group             3.761194       0.000000
            special_group            3.333333       0.000000
     low    normal_group             3.881620       0.043614
            special_group            3.821138       0.044715
     medium normal_group             3.901734       0.056840
            special_group            3.853047       0.053763

                           promotion_last_5years
left salary dep
0    high   normal_group                0.063000
            special_group               0.058065
     low    normal_group                0.010385
            special_group               0.008631
     medium normal_group                0.031322
            special_group               0.050826
```

```
1    high   normal_group                0.000000
            special_group               0.000000
       low    normal_group              0.007269
            special_group               0.000000
       medium normal_group              0.004817
            special_group               0.000000
```

[49]:
```
df=pd.read_csv('HR_Employee_Data.csv')
df["satisfaction_level"] = df["satisfaction_level"].apply(lambda x: x.
 ↪replace('%', '')).astype('float') / 100
df["last_evaluation"] = df["last_evaluation"].apply(lambda x: x.replace('%',␣
 ↪'')).astype('float64') / 100
df["left"] = df["left"].astype("category")

dep = ['RandD',
 'accounting',
 'hr']

df1 = df.copy()


df1['dep'] = df1['Department'].apply(lambda x:"special_group" if x in dep else␣
 ↪"normal_group")
df4 = df1.groupby(["left","salary","dep"]).mean()

fig, axes = plt.subplots(nrows = 1, ncols=2, figsize=(20,10))

status = ["Stayed", "Left"]

for i, statu in enumerate(status):

    high = np.array(df4.iloc[0+6*i:2+6*i]["satisfaction_level"])
    median = df4.iloc[4+6*i:6+6*i]["satisfaction_level"]
    low = df4.iloc[2+6*i:4+6*i]["satisfaction_level"]


    w = .2 # Use 25% of each slot (day of week) per bar, leaving 25% for spacing

    axes[i].bar(np.arange(2)-w, low,  width=w, color='#FFEDA0', label="low")
    axes[i].bar(np.arange(2),   median,  width=w, color='#FEB24C', label="med")
    axes[i].bar(np.arange(2)+w, high, width=w, color='#F03B20', label="high")
    axes[i].spines['top'].set_visible(False)
    axes[i].spines['right'].set_visible(False)
    axes[i].set_title(f'{statu} Employee Satisfaction Level',
                      loc = "center",size=22,weight="bold",c='black')
#     axes[i].spines['left'].set_visible(False)
```

```
    axes[i].set_yticks([0,1])
    axes[i].set_yticklabels(['0','1'])
    axes[i].set_xticks([0,1])
    axes[i].set_xticklabels(['Normal Departments','Special Departments'])



axes[0].legend(bbox_to_anchor=(1.4, -0.1),ncol=3,prop={'size': 16})
axes[0].set_xlabel("Stayed",fontsize=12)
axes[1].set_xlabel("Left",fontsize=12)



axes[1].annotate('0.465',
             (0.14, 0.49),
             c='black',
             size=12, weight='bold')
axes[1].annotate('0.306',
             (1.14, 0.33),
             c='black',
             size=12, weight='bold')
axes[0].annotate('0.652',
             (1.138, 0.68),
             c='black',
             size=12, weight='bold')
# fig.savefig('./Employee Satisfaction Level.png')
plt.show()
```



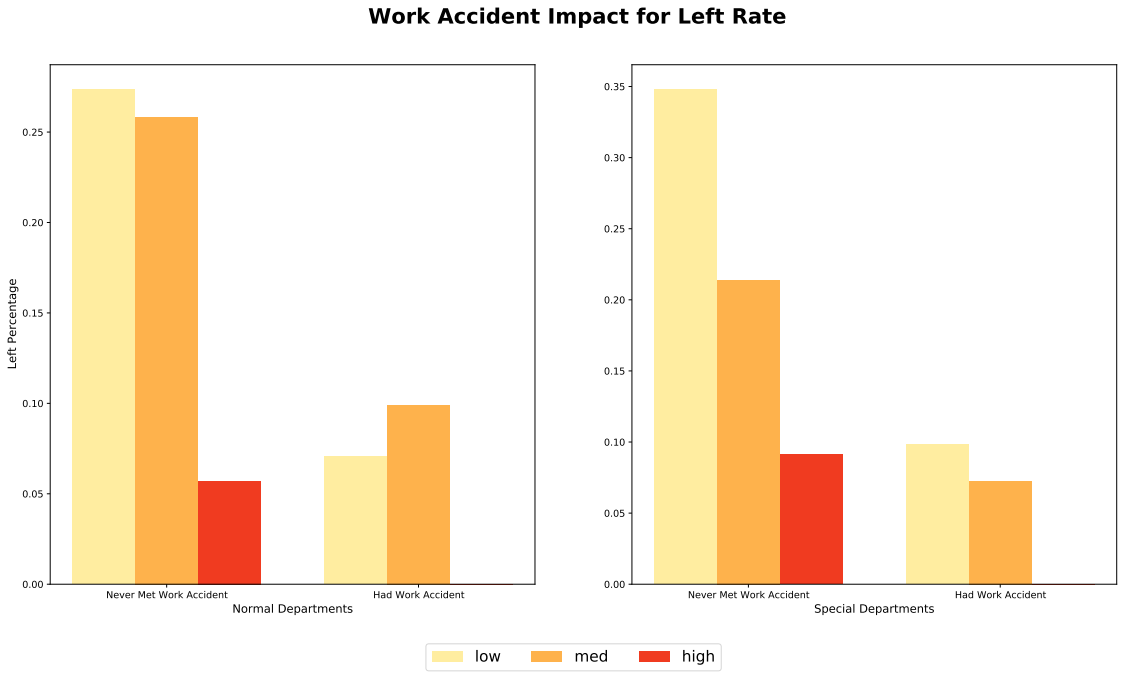
4 Work Accident:

In both of the groups, people who had work accident tend to be more willing to stay in the company than peoople who didn't have accidents in their work.

We believe the current stratgies/benefit the company provides to the employees who had work accidents is good enough to encourage employees to stay, and thus the company may consider to extend the simliar strategies to those employees who don't have work accidents too.

[50]:
```
df=pd.read_csv('HR_Employee_Data.csv')
tab=df.copy()
tab.loc[tab['Department'].isin(['RandD','accounting','hr']),'Department']=0
tab.loc[tab['Department'].
 ↪isin(['IT','marketing','sales','support','technical','management','product_mng']),'Departme
work_ac=tab.groupby(['Department','salary','Work_accident'])['left'].sum()/tab.
 ↪groupby(['Department','salary','Work_accident'])['left'].count()


fig,ax=plt.subplots(figsize=(20,10),ncols=2,nrows=1)
w = .25

high = np.array([0.056995, 0.000000])
med  = np.array([0.258264, 0.099099])
low  = np.array([0.273552,0.070588])

ax[0].bar(np.arange(2)-w, low,  width=w, label="low",color='#FFEDA0')
ax[0].bar(np.arange(2),   med,  width=w, label="med",color='#FEB24C')
ax[0].bar(np.arange(2)+w, high, width=w, label="high",color='#F03B20')
ax[0].set_xticks(np.arange(2))
ax[0].set_xticklabels(['Never Met Work Accident','Had Work Accident'])
ax[0].text(0.8, 0.31, 'Work Accident Impact for Left␣
 ↪Rate',weight="bold",fontdict={'fontsize': 22})
ax[0].set_xlabel("Normal Departments",fontsize=12)
ax[0].set_ylabel("Left Percentage",fontsize=12)

high = np.array([0.091047,0.000000])
med  = np.array([0.213951,0.072727])
low  = np.array([0.347925,0.098089])

ax[1].bar(np.arange(2)-w, low,  width=w, label="low",color='#FFEDA0')
ax[1].bar(np.arange(2),   med,  width=w, label="med",color='#FEB24C')
ax[1].bar(np.arange(2)+w, high, width=w, label="high",color='#F03B20')
ax[1].set_xticks(np.arange(2))
ax[1].set_xticklabels(['Never Met Work Accident','Had Work Accident'])
```

```
ax[1].set_xlabel("Special Departments",fontsize=12)

ax[0].legend(bbox_to_anchor=(1.4, -0.1),ncol=3,prop={'size': 16})

# fig.savefig('./Work Accident Impact for Left Rate.png')
plt.show()
```



5 Average Monthly Hours

For the average monthly hours. we can see if we'd like to improve the satisfaction rate of our employee or say retain our employee, we need to lower the working hours for median and low paying staff and increase the working our of high paying employee. In general, give them more sense of achievement in this job.

[38]:
```
df5 = df1.groupby(["salary","left"]).mean()
df5
# df5.iloc[4:6]["average_montly_hours"]
```

[38]:

		satisfaction_level	last_evaluation	number_project	\
salary	left				
high	0	0.651879	0.708606	3.789610	
	1	0.434512	0.644024	3.451220	
low	0	0.668103	0.717028	3.784603	
	1	0.441248	0.716994	3.836096	
medium	0	0.668875	0.715461	3.788068	

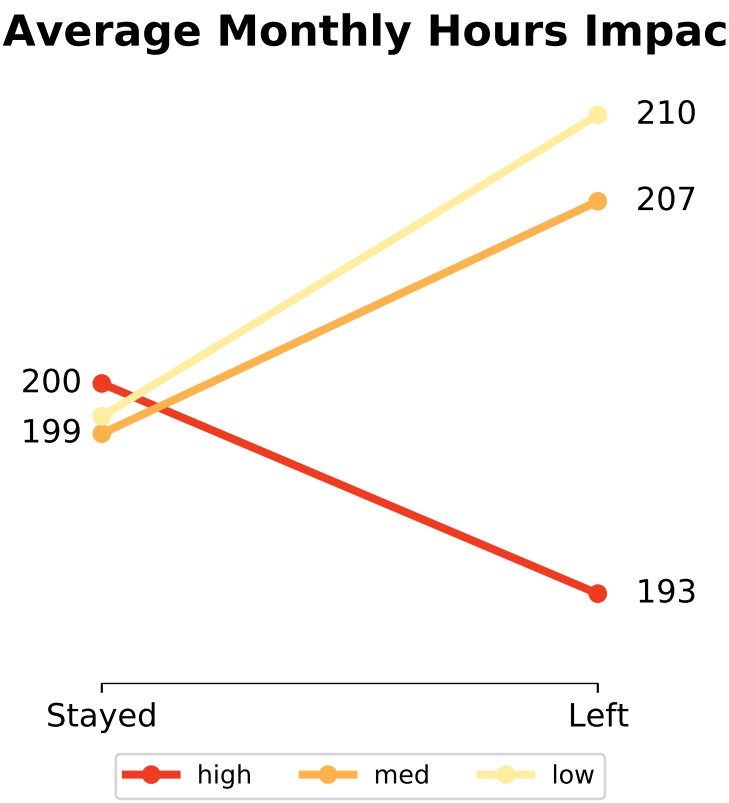
```
       1               0.438550         0.724571        3.912680

              average_montly_hours  time_spend_company  Work_accident  \
salary left
high   0                200.348052            3.693506       0.166234
       1                193.097561            3.682927       0.000000
low    0                198.617418            3.253888       0.183709
       1                206.631215            3.874770       0.043738
medium 0                199.214272            3.435952       0.168259
       1                209.610478            3.891420       0.056188

              promotion_last_5years
salary left
high   0                   0.062338
       1                   0.000000
low    0                   0.010109
       1                   0.006446
medium 0                   0.034315
       1                   0.003797
```

[51]:
```
scores = {}
# df5.iloc[0:2]["satisfaction_level"]
scores['high'] = df5.iloc[0:2]["average_montly_hours"].to_list()
scores['median'] = df5.iloc[2:4]["average_montly_hours"].to_list()
scores['low'] = df5.iloc[4:6]["average_montly_hours"].to_list()

sl=['high','med','low']


fig, ax = plt.subplots(figsize=(4,4))

# Let's use 0 as the left-hand side and 1 as the right-hand side
# (below we will set labels to before-covid for 0 and after-covid for 1)
ax.set_xlim(0-.1,1+.1)
# ax.set_ylim(0.3,1) #match ylim with the range of values
ax.set_ylim(190,210)
# color = '#FFEDA0'
# Draw lines and text associated with scores
wordcolor='black'
i=0
for key in scores.keys():
    a,b = scores[key]
    print(scores[key])
#     a = a/400
#     b=b/400
    color = '#FFEDA0' #diverging
    if key=='high':
```

```
        color = '#F03B20' #highlight scores from Shan using a bright color
    elif key=='median':
        color = '#FEB24C'
    ax.plot([0,1], [a,b], 'o-', lw=3, c=color,label=sl[i]) #line plot with␣
↪x=[0,1] y=[a,b] for each key
    if key != "low":
        ax.text(0-.04, a, f"{a:.0f}", color=wordcolor,
            horizontalalignment='right', verticalalignment='center', size=12)␣
↪#print value of a

    ax.text(1+.2, b, f"{b:.0f}", color=wordcolor,
            horizontalalignment='right', verticalalignment='center', size=12)
    i=i+1

# a,b= scores['low']
# ax.text(1+.04, b, f"{b:.0f}", color=wordcolor,horizontalalignment='left',
#         verticalalignment='center', size=12)

# a,b= scores['median']
# ax.text(1+.04, b-0.5, f"{b:.0f}", color=wordcolor,horizontalalignment='left',
#         verticalalignment='center', size=12)

#print value of b
#     ax.text(0+.42, 0.92, "Survey category | Percent favorable", color=color,
#             horizontalalignment='right', verticalalignment='center', size=12)␣
↪#print names
#     ax.text(0+.27, 0.3-0.05, "Survey year", color=color,
#             horizontalalignment='right', verticalalignment='center', size=12)␣
↪#print
# Make the axes look right
# ax.set_title("Employee feedback over time", size=14)
ax.spines['bottom'].set_bounds(0, 1)#  0 1
ax.set_xticks([0,1])
ax.set_xticklabels(['Stayed','Left'], size=12,color='black')
ax.set_yticks([])#

# Only show the bottom axis
ax.spines['left'].set_visible(False)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_linewidth(.5)


# annotate
ax.annotate("Average Monthly Hours Impact", (0-0.2,212 ), size=16,␣
↪color='black', annotation_clip=False,weight='bold')
```

```
ax.legend(bbox_to_anchor=(0.95, -0.1),ncol=3,prop={'size': 9.5})

# fig.savefig('./Average Monthly Hours Impact.png')
#title
plt.show()
```

```
[200.34805194805196, 193.09756097560975]
[198.61741835147745, 206.63121546961327]
[199.2142717878729, 209.61047835990888]
```



6 Number Project

#explain Two plots

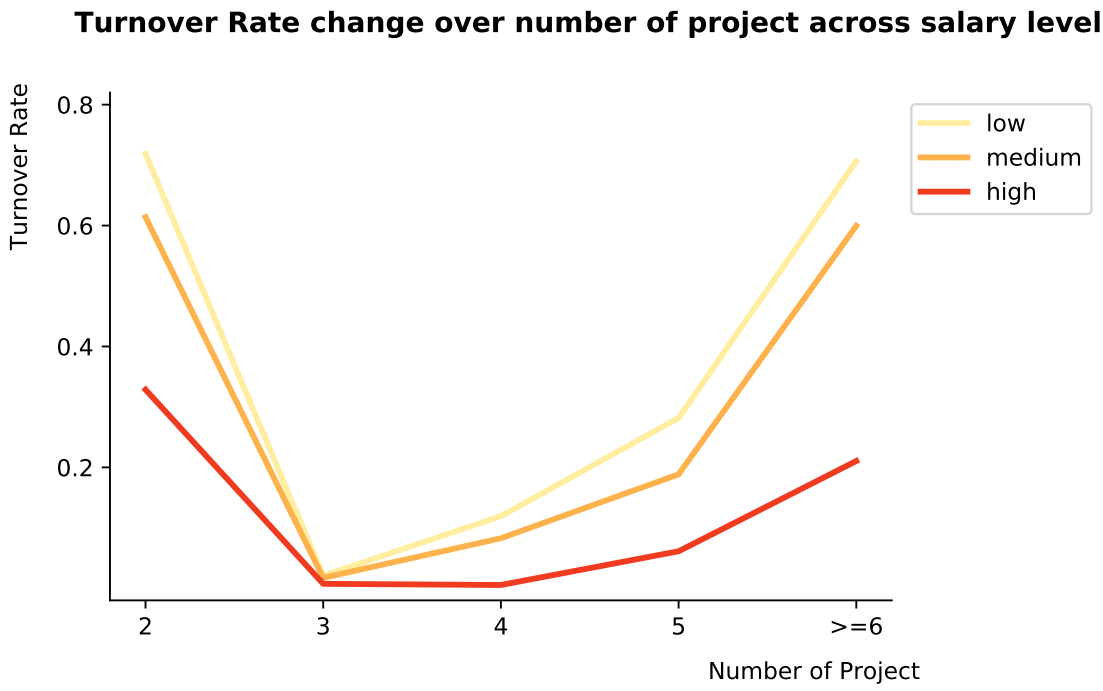
First: Overview of **number of project** against different salary level

Second: Abnormal group explanation; Observe a difference of turnover rate between abnormal group and normal group in low and medium salary level The value is

$$Turnover_{medium,i,j} - Turnover_{low,i,j}$$

$$i : number\ of\ project, j : group$$

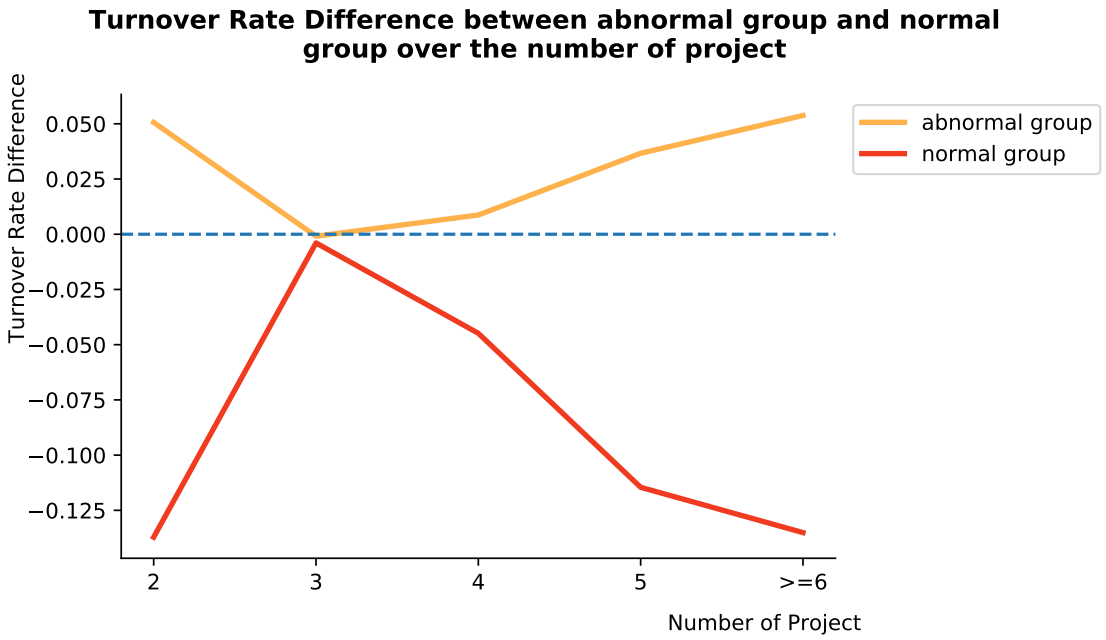
[52]:
```
#code
df.loc[df['number_project']>=6,'number_project']=6
tab1=df.groupby(['salary','number_project'])['left'].sum()/df.
 ↪groupby(['salary','number_project'])['left'].count()
tab1=tab1.reset_index()
fig, ax = plt.subplots(figsize=(6,4))
xlist=[2,3,4,5,'>=6']
slist=['low','medium','high']
clist=[
'#ffeda0',
'#feb24c',
'#f03b20']
for i in range(3):
    tab=tab1[tab1['salary']==slist[i]]
    ax.plot(xlist, tab.left,label=slist[i],c=clist[i],lw=2.5)
ax.set_ylim(-0.02,0.82)
ax.set_yticks(np.arange(0.2,1,0.2))
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xlabel('Number of Project',size=10)
ax.set_ylabel('Turnover Rate',size=10)
ax.yaxis.set_label_coords(-0.1,0.85)
ax.xaxis.set_label_coords(0.9,-0.12)
ax.text(-0.4,0.92,'Turnover Rate change over number of project across salary␣
 ↪level',weight='bold',size=12)
plt.legend(bbox_to_anchor=(1.01, 1))
# fig.savefig('./Turnover Rate change over number of project across salary leve.
 ↪png')
plt.show()
```



As normal group presents, the turnover rate difference between the medium salary level and the low level should be no more than 0 generally. In contrast, that for abnormal group is over 0. When taking the number of project into consideration, we could see that when staff work on three projects, there is nearly no difference on the turnover rate. However, other than 3, the number of project changes the turnover trend in an opposite way between groups. In abnormal group, with increasing number of project, the turnover rate for staff with medium salary level is simultaneously increasing, which means the staff with medium salary care more about the number of project(i.e. the work intensity). In normal group, it decreases oppositely.

[53]:
```
tab5=df.copy()
# tab5.loc[tab5['Department'].
 ↪isin(['RandD','accounting','hr','management','product_mng']),'Department']=0
# tab5.loc[tab5['Department'].
 ↪isin(['IT','marketing','sales','support','technical']),'Department']=1
tab5.loc[tab5['Department'].isin(['RandD','accounting','hr']),'Department']=0
tab5.loc[tab5['Department'].
 ↪isin(['IT','marketing','sales','support','technical','management','product_mng']),'Departme
tab5_1=tab5.groupby(['Department','salary','number_project'])['left'].sum()/
 ↪tab5.groupby(['Department','salary','number_project'])['left'].count()
tab5_1=tab5_1.reset_index()
tab5_1
# Turnover rate difference (against the number of project)between two groups we␣
 ↪classified across different salary level
fig,ax=plt.subplots(figsize=(6,4))
```

```
xlist=[2,3,4,5,'>=6']
slist=['medium','low']
clist=['#feb24c',
'#f03b20']
label=['abnormal group','normal group']
for i in range(2):
    tab5_2=tab5_1[(tab5_1['Department']==i)&(tab5_1['salary']==slist[0])].
 ↪reset_index(drop=True)
    tab5_3=tab5_1[(tab5_1['Department']==i)&(tab5_1['salary']==slist[1])].
 ↪reset_index(drop=True)
    ax.plot(xlist,tab5_2['left']-tab5_3['left'],c=clist[i],label=label[i],lw=2.
 ↪5)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xlabel('Number of Project')
ax.set_ylabel('Turnover Rate Difference')
ax.yaxis.set_label_coords(-0.13,0.75)
ax.xaxis.set_label_coords(0.9,-0.12)
ax.text(-0.4,0.08,'''Turnover Rate Difference between abnormal group and normal
                        group over the number of␣
 ↪project''',weight='bold',size=12)
ax.axhline(y=0,linestyle='--')
plt.legend(bbox_to_anchor=(1.01, 1))
# fig.savefig('./Turnover Rate Difference between abnormal.png')
plt.show()
```



7 Commuting time

#explain

Two plots

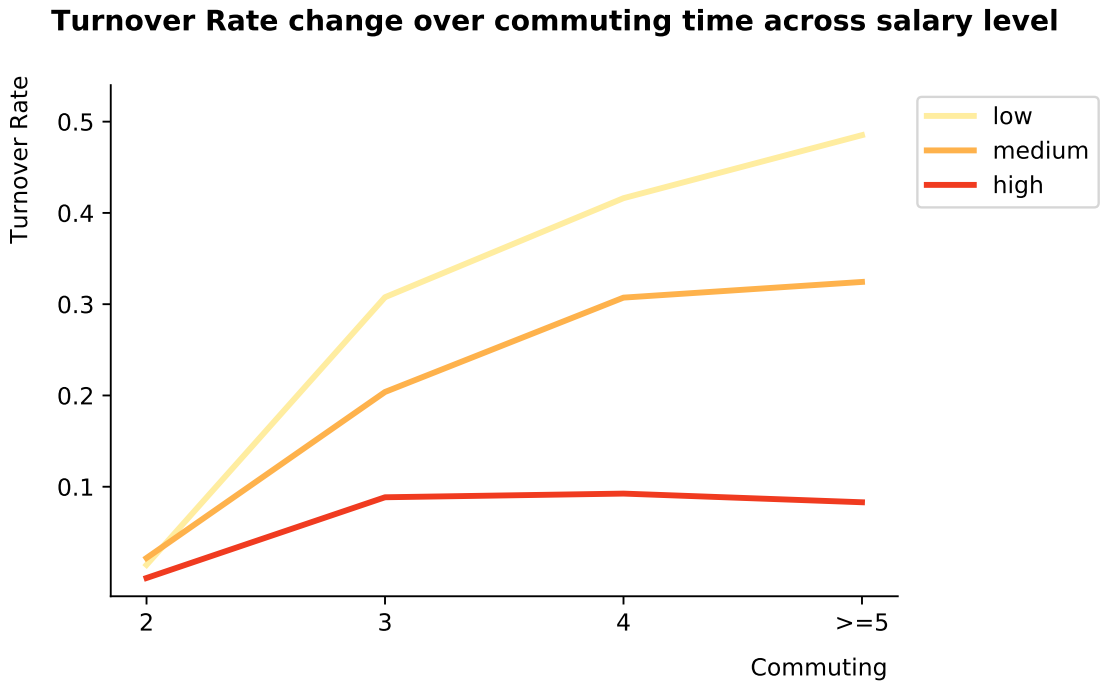
First: Overview of **commuting time** against different salary level

Second: Abnormal group explanation; Observe a difference of turnover rate between abnormal group and normal group in low and medium salary level. The value is

$$Turnover_{medium,i,j} - Turnover_{low,i,j}$$

$$i : commuting\ time\ , j : group$$

[54]:
```
df.loc[df['time_spend_company']>=5,'time_spend_company']=5
tab2=df.groupby(['salary','time_spend_company'])['left'].sum()/df.
 ↪groupby(['salary','time_spend_company'])['left'].count()
tab2=tab2.reset_index()
fig, ax = plt.subplots(figsize=(6,4))
xlist=[2,3,4,'>=5']
slist=['low','medium','high']
clist=[
'#ffeda0',
'#feb24c',
'#f03b20']
for i in range(3):
    tab=tab2[tab2['salary']==slist[i]]
    ax.plot(xlist, tab.left,label=slist[i],c=clist[i],lw=2.5)
ax.set_ylim(-0.02,0.54)
ax.set_yticks(np.arange(0.1,0.6,0.1))
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xlabel('Commuting',size=10)
ax.set_ylabel('Turnover Rate',size=10)
ax.yaxis.set_label_coords(-0.1,0.85)
ax.xaxis.set_label_coords(0.9,-0.12)
ax.text(-0.4,0.6,'Turnover Rate change over commuting time across salary␣
 ↪level',weight='bold',size=12)
plt.legend(bbox_to_anchor=(1.01, 1))
# fig.savefig('./Turnover Rate change over commuting time across.png')
plt.show()
```



When considering commuting time, we could see that when staff work on three projects, there is nearly no difference on the turnover rate when commuting time is 2 hours. However, changing it from 3 to 4 hours, the turnover rate difference presents opposite trend in the abnormal group and normal group. In the abnormal group, the rate difference is increasing, which means the staff with medium salary care more about the commuting time. In contrast, the staff with low salary care more about the commuting time in the normal group.

[55]:
```
#code
tab6=df.copy()
tab6.loc[tab6['Department'].isin(['RandD','accounting','hr']),'Department']=0
tab6.loc[tab6['Department'].
 ↪isin(['IT','marketing','sales','support','technical','management','product_mng']),'Departme
tab6_1=tab6.groupby(['Department','salary','time_spend_company'])['left'].sum()/
 ↪tab6.groupby(['Department','salary','time_spend_company'])['left'].count()
tab6_1=tab6_1.reset_index()
fig,ax=plt.subplots(figsize=(6,4))
xlist=[2,3,4,'>=5']
slist=['medium','low']
clist=[
'#feb24c',
'#f03b20']
label=['abnormal group','normal group']
for i in range(2):
```

```
    tab6_2=tab6_1[(tab6_1['Department']==i)&(tab6_1['salary']==slist[0])].
↪reset_index(drop=True)
    tab6_3=tab6_1[(tab6_1['Department']==i)&(tab6_1['salary']==slist[1])].
↪reset_index(drop=True)
    ax.plot(xlist,tab6_2['left']-tab6_3['left'],c=clist[i],label=label[i],lw=2.
↪5)
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.set_xlabel('Time spent on commuting')
ax.set_ylabel('Turnover Rate Difference')
ax.text(-0.4,0.12,'''Turnover Rate Difference between abnormal group and normal
                     group over the commuting time''',weight='bold',size=12)
ax.axhline(y=0,linestyle='--')
plt.legend(bbox_to_anchor=(1.01, 1))
# fig.savefig('./Turnover Rate Difference between abnormal group and normal␣
↪group over the commuting time.jpg')
plt.show()
```