```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import linregress

mass_nucleon = (0.938272+0.9406)/2
alpha_fine = 1/137
initial_correction = 1.0

df = []

with open('data.dat', 'r') as file:
    for line in file:
        numbers = line.split()
        df.append([float(num) for num in numbers])
df = pd.DataFrame(df)
df.columns = ["Z","A","E0","ThetaDeg","nu","cross","error","dataSet"]
added = pd.read_excel("DATA_55.1_145_deg(add_up).xlsx")
added.columns =
["Z","A","E0","ThetaDeg","nu","cross","error","dataSet"]
df = pd.concat([df,added],ignore_index=True)
# df = pd.read_excel("data_raw.xlsx")
# df

df["R"]=1.1*(df["A"])**(1/3)+0.86*(df["A"])**(-1/3)
df["Veff"]=0.775*(3/2)*alpha_fine*(df["Z"]-1)/df["R"]
df["Eeff"]=df["E0"]+df["Veff"]
df["ThetaRad"]=df["ThetaDeg"]*np.pi/180
df["sin2(T/2)"]=(np.sin(df["ThetaRad"]/2))**2
df["cos2(T/2)"]=(np.cos(df["ThetaRad"]/2))**2
df["tan2(T/2)"]=(np.tan(df["ThetaRad"]/2))**2
df["Eprime"]=df["E0"]-df["nu"]
df["Eprime_eff"]=df["Eprime"]+df["Veff"]


df["Q2"]=4*df["E0"]*(df["Eprime"])*df["sin2(T/2)"]
df["Q2eff"]=4*df["Eeff"]*df["Eprime_eff"]*df["sin2(T/2)"]



# df["q3momt_squared"]=df["nu"]**2+df["Q2"]
df["q3momt_squared"]=df["nu"]**2+df["Q2eff"]

df["q3momt"]=np.sqrt(df["q3momt_squared"])

# df["W2"]=mass_nucleon**2+2*mass_nucleon*df["nu"]-df["Q2"]
df["W2"]=mass_nucleon**2+2*mass_nucleon*df["nu"]-df["Q2eff"]

# df["W"]=np.sqrt(df["W2"])

# df["epsilon"]=1/(1+2*(1+(df["nu"]**2)/df["Q2"])*df["tan2(T/2)"])
```

```
df["epsilon"]=1/(1+2*(1+(df["nu"]**2)/df["Q2eff"])*df["tan2(T/2)"])





## QUESTIONS!!!!!
#
df["gamma"]=alpha_fine*df["Eprime"]*(df["W2"]-mass_nucleon**2)/((4*((n
p.pi)**2)*df["Q2"]*mass_nucleon*df["E0"])*(1-df["epsilon"]))
df["gamma"]=alpha_fine*df["Eprime_eff"]*(df["W2"]-mass_nucleon**2)/
((4*((np.pi)**2)*df["Q2eff"]*mass_nucleon*df["E0"])*(1-df["epsilon"]))

df["Sig_R"]=df["cross"]/df["gamma"]
df["D_sig_R"]=df["error"]/df["gamma"]

df["Sig_mott"]=4*(alpha_fine**2)*(df["Eprime"]**2)*df["cos2(T/2)"]/
(df["Q2"]**2)
df["Sig_mott_eff"]=df["Sig_mott"]*df["E0"]/df["Eeff"]

#
df["H"]=(df["q3momt_squared"]**2)/(4*(alpha_fine**2)*(df["Eprime"]**2)
*(df["cos2(T/2)"]+2*(df["q3momt_squared"]/df["Q2"])*df["sin2(T/2)"]))
df["H"]=(df["q3momt_squared"]**2)/(4*(alpha_fine**2)*(df["Eprime_eff"]
**2)*(df["cos2(T/2)"]+2*(df["q3momt_squared"]/
df["Q2eff"])*df["sin2(T/2)"]))

df["Hstar_Sig(nb)"]=initial_correction*df["H"]*df["cross"]
df["Hstar_error(nb)"]=initial_correction*df["H"]*df["error"]
df["Hstar_Sig(GeV)"]=df["Hstar_Sig(nb)"]/((0.1973269**2)*10000000)
df["Hstar_error(GeV)"]=df["Hstar_error(nb)"]/((0.1973269**2)*10000000)


df = df.sort_values(by="Q2")
# bins =
[0.003,0.015,0.039,0.060,0.086,0.120,0.165,0.240,0.330,0.470,0.660,0.9
75,1.585,2.285,3.150,4.600,7.250]
'''
Q2=0.012 (0.05- 0.028) (no E04-001 data)

Q2 = 0.03 (0.028-0.039) (no E04-001 data) *

Q2= 0.05 (0.039-0.060) 1.2 GeV 10.8 deg *
Q2= 0.07 (0.060-0.085) 1.2 GeV 13 deg *
Q2=0.10 (0.085-0.120) 1.2 GeV 16 deg *
Q2=0.14 (0.120-0.165) 1.2 GeV 19 deg *
Q2=0.19 (0.16-0.248) 1.2 GeV 22 deg *
Q2=0.26 (0.248-0.280) no E04-001 data) *
Q2=0.30 (0.280-0.338) 1.2 GeV 28 deg *
Q2= 0.37 (0.338-0.470) (no E04-001 data) *
```

```
Q2= 0.57 (0.470-0.660) 1.2 GeV 45 deg, 3.49 GeV 14 deg
Q2 =0.75 (0.660-0.90) 4.63 GeV 10.65 deg. 1.2 GeV 55 deg
Q2=1.0 (0.90-1.2) 1.207 70 deg, 463 GeV 13 deg,2.35 GeV 30 deg, 3.49
GeV 20 deg, 4.63 GeV 16 deg).

'''
bins =
[0.004,0.025,0.042,0.068,0.095,0.12,0.18,0.25,0.31,0.36,0.56,0.83,1.16
,1.46,1.7,2.1,2.5]
# Q2bins = ["0.003~0.015", "0.015~0.039", "0.039~0.060",
"0.060~0.085", "0.085~0.120", "0.120~0.165",
#             "0.165~0.240", "0.240~0.330", "0.330~0.470",
"0.470~0.660", "0.660~0.975", "0.975~1.585",
#             "1.585~2.285", "2.285~3.150", "3.150~4.600",
"4.600~7.250"]
Q2bins=["0.004~0.025","0.025~0.042","0.042~0.068","0.068~0.095","0.095
~0.120","0.120~0.180",

"0.180~0.250","0.250~0.310","0.310~0.360","0.360~0.560","0.560~0.830",
"0.830~1.160",
        "1.160~1.460","1.460~1.700","1.700~2.100","2.100~2.500"]
# Q2center=[0.010, 0.028, 0.050, 0.070, 0.100, 0.140, 0.190, 0.290,
0.370, 0.570, 0.750, 1.200, 1.970, 2.600, 3.700, 5.500]
#
Q2center=[0.012,0.03,0.05,0.07,0.10,0.14,0.19,0.26,0.30,0.37,0.57,0.75
,1.0]
Q2bin_to_Q2center = {
    "0.004~0.025":0.0145,
    "0.025~0.042":0.0335,
    "0.042~0.068":0.055,
    "0.068~0.095":0.0815,
    "0.095~0.120":0.1075,
    "0.120~0.180":0.15,
    "0.180~0.250":0.215,
    "0.250~0.310":0.28,
    "0.310~0.360":0.335,
    "0.360~0.560":0.46,
    "0.560~0.830":0.695,
    "0.830~1.160":0.995,
    "1.160~1.460":1.31,
    "1.460~1.700":1.58,
    "1.700~2.100":1.9,
    "2.100~2.500":2.3
}

df["bin"] = pd.cut(x=df["Q2"], bins=bins, labels=Q2bins, right=True)
df["Q2center"]=df["bin"].map(Q2bin_to_Q2center)
df = df.groupby("bin").apply(lambda x:
x.sort_values(by="W2")).reset_index(drop=True)
```

```
df["RL"]=0
df["RT"]=0

dataSet_to_name = {
    1:"Barreau:1983ht",
    2:"O'Connell:1987ag",
    3:"Sealock:1989nx",
    4:"Baran:1988tw",
    5:"Bagdasaryan:1988hp",
    6:"Dai - HallA:2019da",
    7:"Arrington:1995hs",
    8:"Day:1993md",
    9:"Arrington:1998psnoCC",
    10:"Gaskell:2008",
    11:"Whitney:1974hr",
    12:"AlsamiJan05",
    13:"VaheJun07",
    14:"E139",
    15:"Fomin",
    16:"Yamaguchi73",
    17:"Ryan84",
    18:"Bounin63",
    21:"DATA_55.1_145_deg"
}
Q2bin_to_W2width={
    "0.004~0.025":0.01,
    "0.025~0.042":0.01,
    "0.042~0.068":0.01,
    "0.068~0.095":0.01,
    "0.095~0.120":0.01,
    "0.120~0.180":0.01,
    "0.180~0.250":0.01,
    "0.250~0.310":0.01,
    "0.310~0.360":0.01,
    "0.360~0.560":0.01,
    "0.560~0.830":0.03,
    "0.830~1.160":0.03,
    "1.160~1.460":0.03,
    "1.460~1.700":0.03,
    "1.700~2.100":0.03,
    "2.100~2.500":0.03
}

'''
Coulomb sum rule
Rosenbluth
Negative W_squared
1. sort in Q2 and bin center
    Bin cetering for Q2
    center  low     high
```

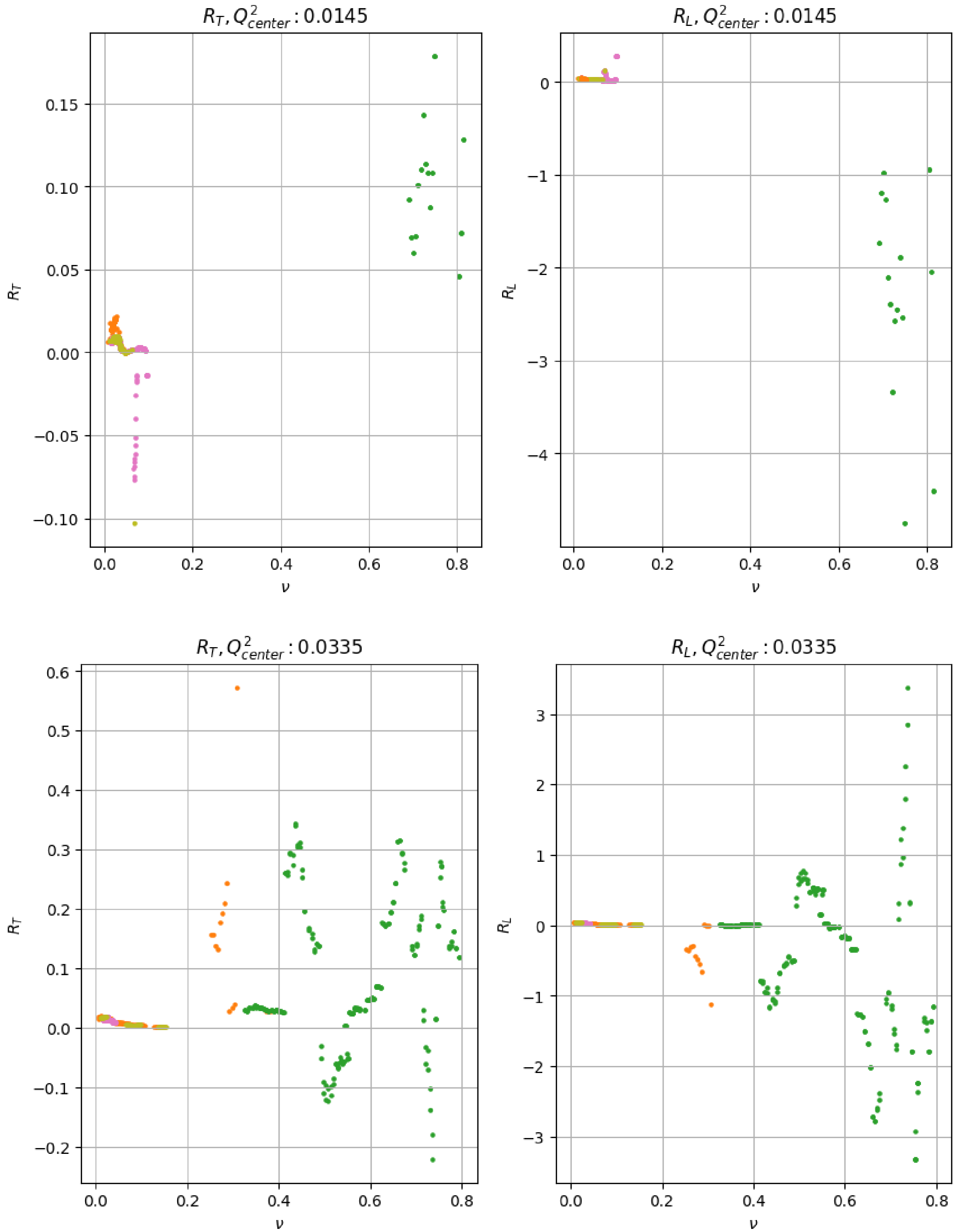
```
    0.010   0.003   0.015
    0.028   0.015   0.039
    0.050   0.039   0.060
    0.070   0.060   0.085
    0.100   0.085   0.120
    0.140   0.120   0.165
    0.190   0.165   0.240
    0.290   0.240   0.330
    0.370   0.330   0.470
    0.570   0.470   0.660
    0.750   0.660   0.975
    1.200   0.975   1.585
    1.970   1.585   2.285
    2.600   2.285   3.150
    3.700   3.150   4.600
    5.500   4.600   7.250

2. for each bin, sort in W2, focus on range 0.925~0.935
    0.925~0.930?
3. Plot RT, RL versus nu



Q2=0.012 (0.05- 0.028) (no E04-001 data)

Q2 = 0.03 (0.028-0.039) (no E04-001 data) *

Q2= 0.05 (0.039-0.060) 1.2 GeV 10.8 deg *
Q2= 0.07 (0.060-0.085) 1.2 GeV 13 deg *
Q2=0.10 (0.085-0.120) 1.2 GeV 16 deg *
Q2=0.14 (0.120-0.165) 1.2 GeV 19 deg *
Q2=0.19 (0.16-0.248) 1.2 GeV 22 deg *
Q2=0.26 (0.248-0.280) no E04-001 data) *
Q2=0.30 (0.280-0.338) 1.2 GeV 28 deg *
Q2= 0.37 (0.338-0.470) (no E04-001 data) *
Q2= 0.57 (0.470-0.660) 1.2 GeV 45 deg, 3.49 GeV 14 deg
Q2 =0.75 (0.660-0.90) 4.63 GeV 10.65 deg. 1.2 GeV 55 deg
Q2=1.0 (0.90-1.2) 1.207 70 deg, 463 GeV 13 deg,2.35 GeV 30 deg, 3.49
GeV 20 deg, 4.63 GeV 16 deg).



'''

# plot regression line
# for bin in Q2bins:
#     picked = df.loc[(0.925<=df["W2"]) & (df["W2"]<=0.930) &
(df["bin"] == bin)]
#     dup = picked[picked.duplicated()]
#     drop_dup = picked.drop_duplicates(keep=False)#remove all
```

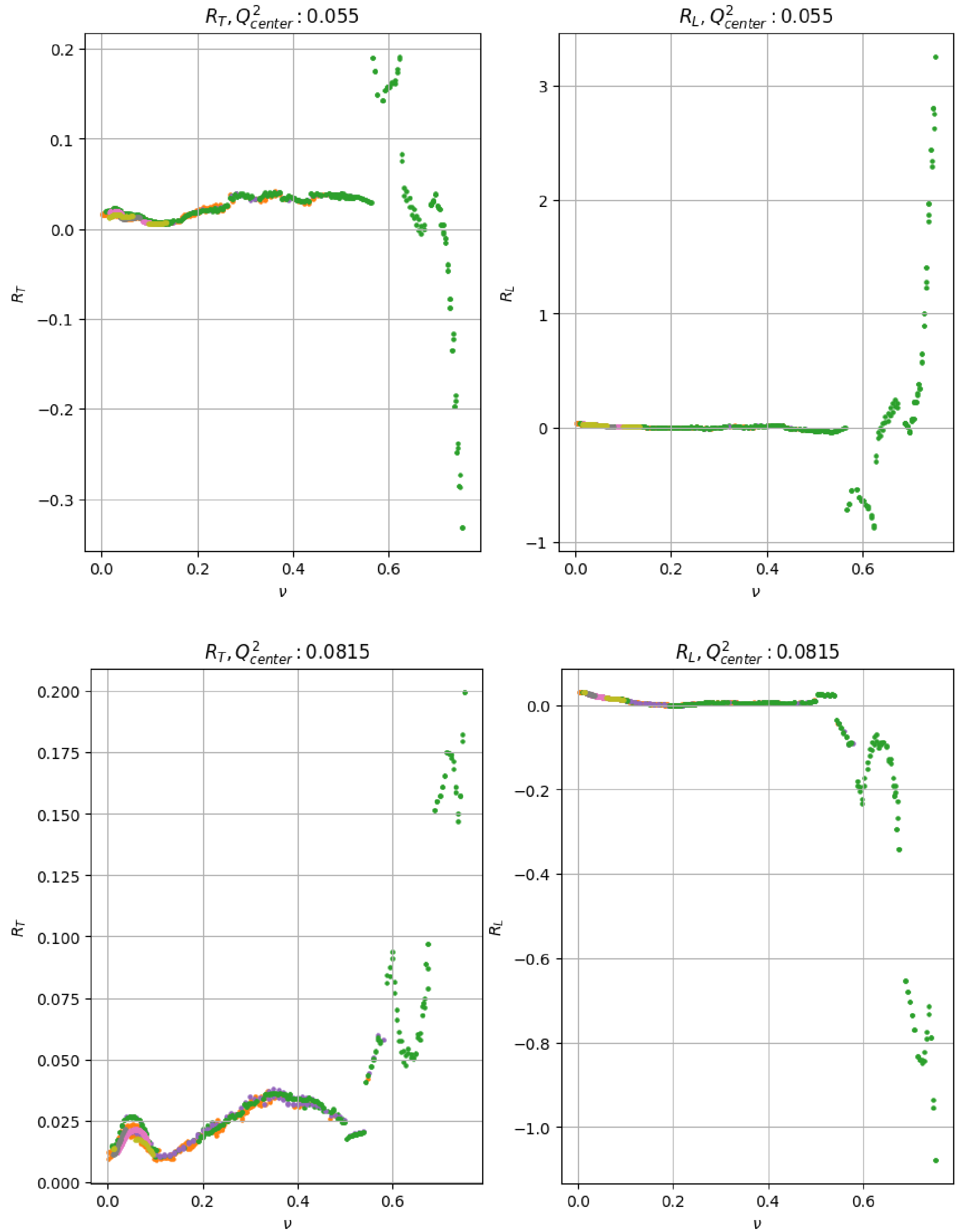
```
duplicates, even the first occurrance
#     plt.figure(figsize=(5,3))
#     x = picked["epsilon"].values
#     y = picked["Hstar_Sig(GeV)"].values

#     if len(x) >= 2:
#         slope, intercept, r_value, p_value, std_err = linregress(x,
y)
#         plt.plot(x, slope*x+intercept, color='orange',
label='y='+str(round(slope,3))+'*x+'+str(round(intercept,3)))
#         plt.legend()
#     if len(dup) > 0:
#         x_drop = drop_dup["epsilon"].values
#         y_drop = drop_dup["Hstar_Sig(GeV)"].values
#
plt.scatter(dup["epsilon"],dup["Hstar_Sig(GeV)"],color="red",s=20)
#         slope, intercept, r_value, p_value, std_err =
linregress(x_drop, y_drop)
#         plt.plot(x_drop, slope*x_drop+intercept, color='green',
label='y='+str(round(slope,3))+'*x+'+str(round(intercept,3)))
#         plt.legend()
#     plt.scatter(x,y,marker="+", s=20)
#     plt.xlabel("$\epsilon$")
#     plt.ylabel("$H^*_\sigma$")
#     plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
#     plt.grid()
#     plt.show()


'''
2 or 4 or 8 MeV W2 bins, 0.91~0.94
error bar
change sig_M_eff
E to E_eff, V to V_eff (3.11), Q2 to Q2_eff
remove 1.2 GeV, 16deg
Coulumb correction, then bin centering correction
RL / RT (correlation matrix to find error bar of ratio)
find 6 plots


for each Q2 bin center:
    sort W2
    loop through nu
    for every nu, using Q2center, find the +-4MeV W2 range

    find RL, RT


Questions:
```

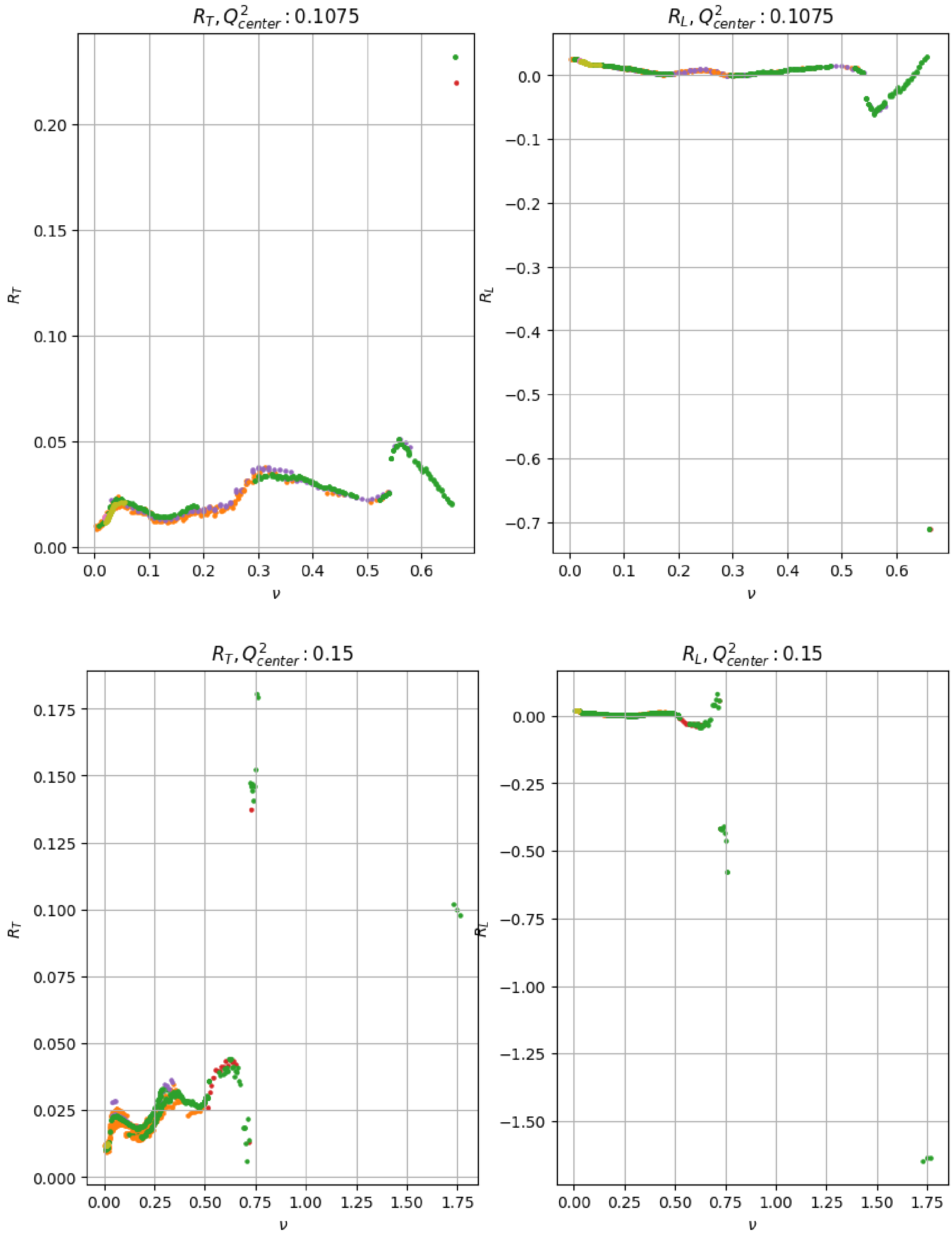
```
gamma using Eprime or E0?
for each Q2 bin:
    loop through nu, for each nu:
        what if there's no enough W2 values to fit 4MeV?
Bin in Q2eff or Q2?

Large fluctuations in greater nu values

'''

for index, row in df.iterrows():
    W2center = mass_nucleon**2+2*mass_nucleon*row["nu"]-
row["Q2center"]**2
    W2width = Q2bin_to_Q2center[row["bin"]]
    picked = df.loc[(df["bin"]==row["bin"]) & ((W2center-
W2width)<=df["W2"]) & (df["W2"]<=(W2center+W2width))]
    # picked = picked.drop_duplicates(keep=False)#remove all
duplicates, even the first occurrance
    picked = picked.drop_duplicates(keep="last")#remove all
duplicates, even the first occurrance

    if len(picked)>=2:
        x = picked["epsilon"].values
        y = picked["Hstar_Sig(GeV)"].values
        try:
            slope, intercept, r_value, p_value, std_err =
linregress(x, y)
            # row["RL"]=slope/1000
            df.at[index, "RL"] = slope/1000
            #Question: q3momt_squared or q3momt_squared_center?
            #
row["RT"]=(2*intercept*row["Q2center"]/row["q3momt_squared"])/1000
            df.at[index, "RT"] =
(2*intercept*row["Q2center"]/row["q3momt_squared"])/1000
            # print("RL,RT:",RL,RT)
        except ValueError:
            print("Value error")
    # else:
    #     print("less than 3")

df.to_excel("data_fit.xlsx", index=False)
# df.to_csv("data_fit.csv", index=False)

for bin_name in Q2bins:
    fig = plt.figure(figsize=(10, 6))
    Q2center = Q2bin_to_Q2center[bin_name]

    # bin = df.loc[df["bin"]==bin_name]
    bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!
=0)]
```

```
    plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
    for i in range(0,22):
        plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
        # plt.scatter(bin["nu"],bin["RT"],s=5)
    # plt.xlim(0, 0.4)
    # plt.ylim(0, 0.05)
    plt.xlabel("$\\nu$")
    plt.ylabel("$R_T$")
    # plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
    plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
    plt.grid()

    # Plot your data here
    # Create the second plot on the right
    plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
    # Plot your data here
    for i in range(0,22):
        plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
        # plt.scatter(bin["nu"],bin["RT"],s=5)
    # plt.xlim(0, 0.4)
    # plt.ylim(0, 0.05)
    plt.xlabel("$\\nu$")
    plt.ylabel("$R_L$")
    # plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
    plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
    plt.grid()

    # Adjust spacing between subplots
    # plt.tight_layout()
    # plt.legend()

    # Display the plots
    plt.show()


    # plt.scatter((bin.loc[(bin["dataSet"] == 2)])
["nu"],bin.loc[(bin["dataSet"] == 2)]["RT"],s=5)


    # plt.legend()


# for bin_name in Q2bins:
#     # picked = df.loc[(0.925<=df["W2"]) & (df["W2"]<=0.930) &
```

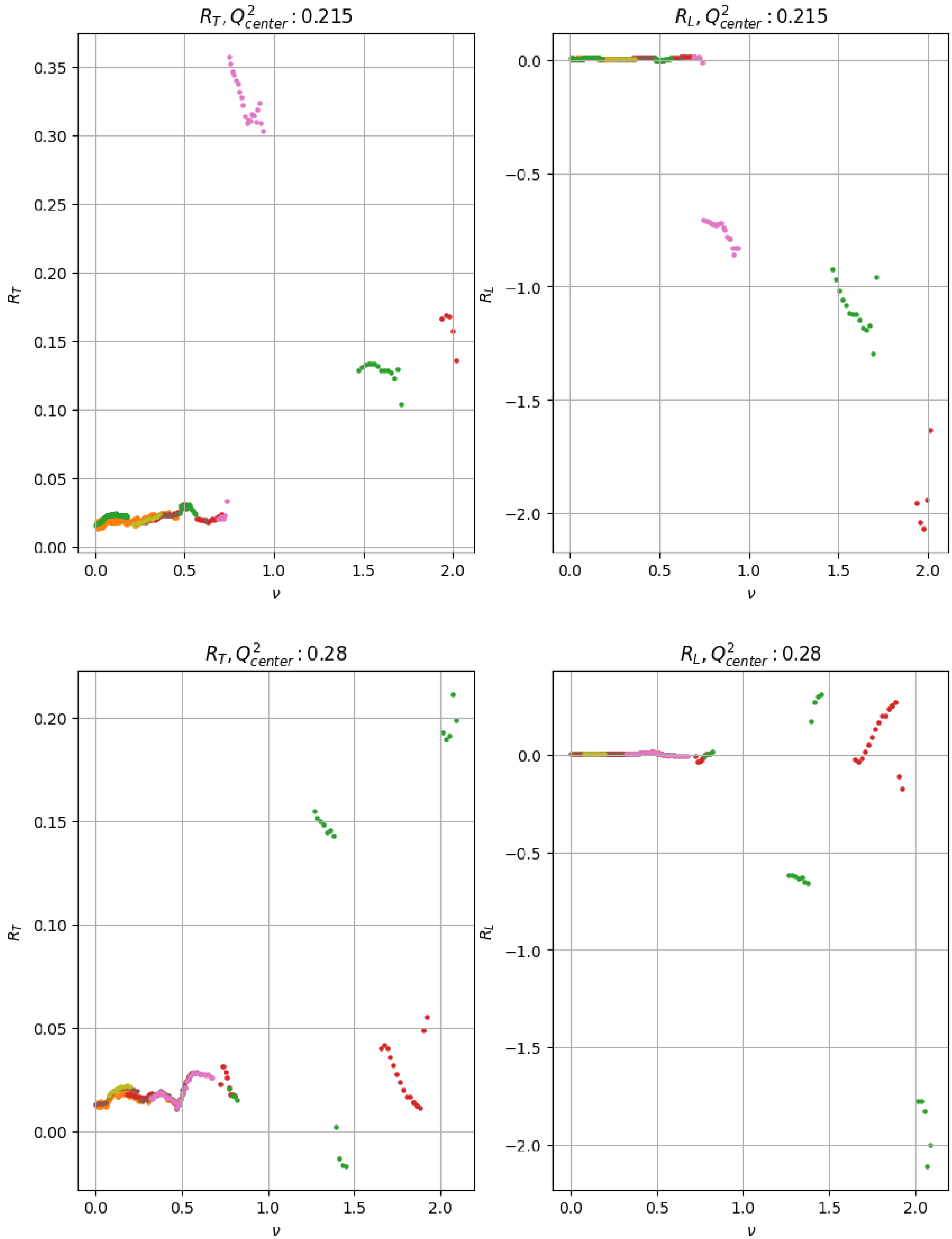
```
(df["bin"] == bin)]
#     bin = df.loc[df["bin"]==bin_name]
#     # looping through each row (all of whom has a nu value)
#     for index, row in bin.iterrows():
#         # find W2center
#         W2center = mass_nucleon**2+2*mass_nucleon*row["nu"]-
row["Q2center"]**2
#         # print("W2center:",W2center)
#         # find W2center bins +-4MeV
#         picked = bin.loc[((W2center-0.004)<=bin["W2"]) &
(bin["W2"]<=(W2center+0.004))]
#         picked = picked.drop_duplicates(keep=False)#remove all
duplicates, even the first occurrance
#         if len(picked)>1:
#             x = picked["epsilon"].values
#             y = picked["Hstar_Sig(GeV)"].values
#             # print("x:",x)
#             # print("y:",y)
#             RL = 0
#             RT = 0
#             try:
#                 slope, intercept, r_value, p_value, std_err =
linregress(x, y)
#                 RL = slope
#                 RT =
2*intercept*row["Q2center"]/row["q3momt_squared"]
#                 print("RL,RT:",RL,RT)

#             except ValueError:
#                 print("Value error")
```


$R_T, Q^2_{center}: 0.0145$
R_T
ν
$R_L, Q^2_{center}: 0.0145$
R_L
ν
$R_T, Q^2_{center}: 0.0335$
R_T
ν
$R_L, Q^2_{center}: 0.0335$
R_L
ν

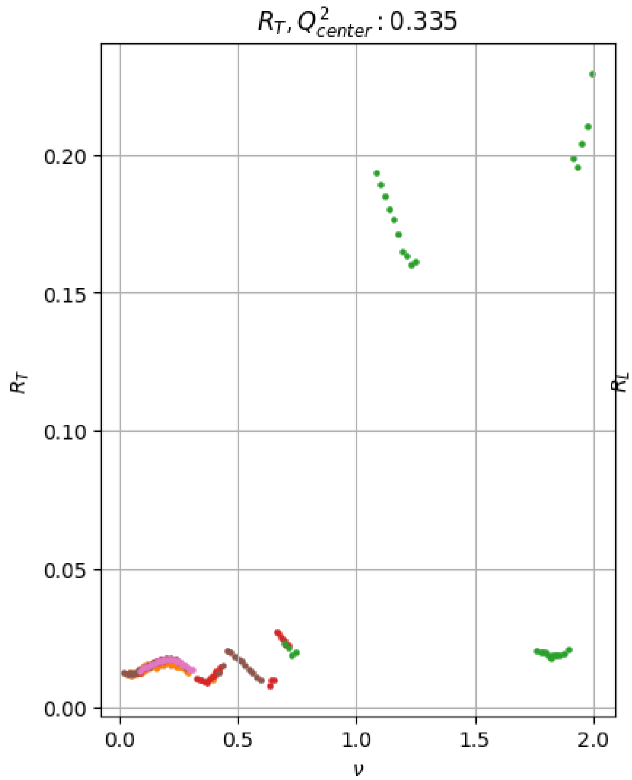

$R_T, Q^2_{center}: 0.055$
$R_L, Q^2_{center}: 0.055$
$R_T, Q^2_{center}: 0.0815$
$R_L, Q^2_{center}: 0.0815$
R_T
R_L
ν

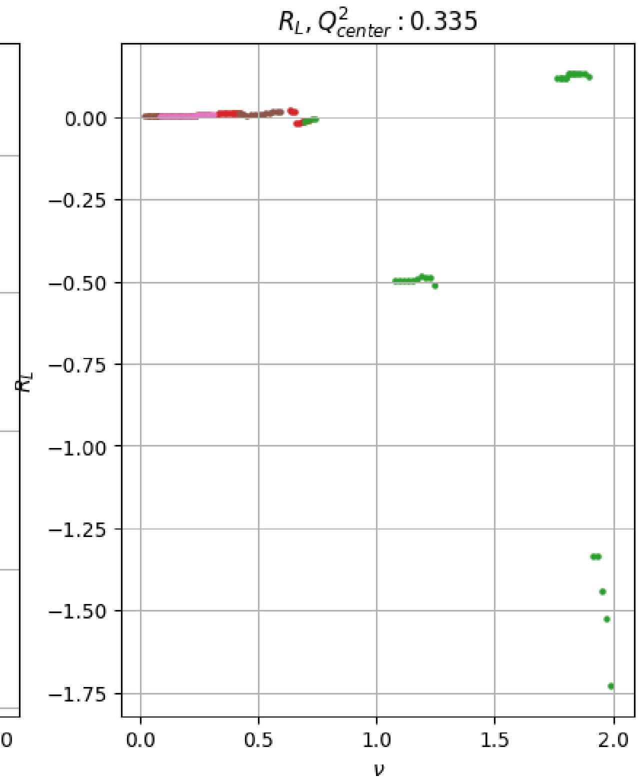


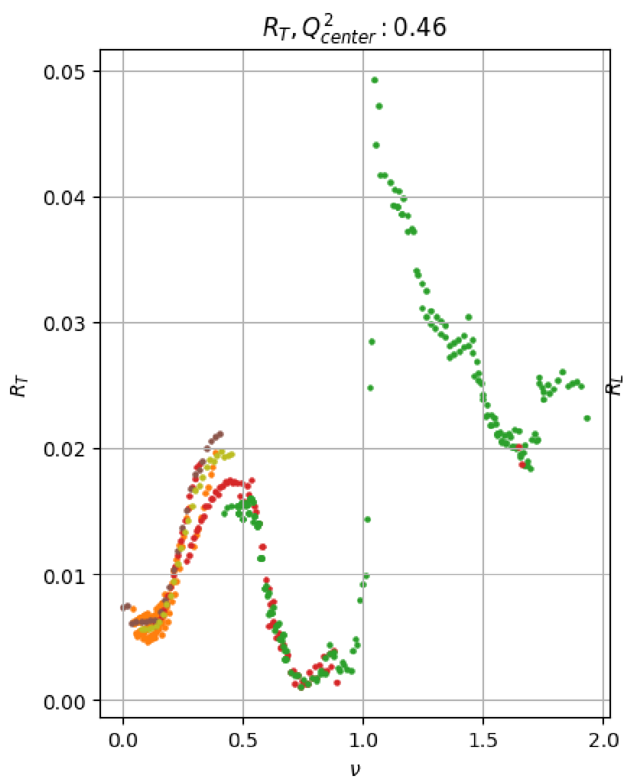
$R_T, Q^2_{center}: 0.1075$
R_T
ν
$R_L, Q^2_{center}: 0.1075$
R_L
ν
$R_T, Q^2_{center}: 0.15$
R_T
ν
$R_L, Q^2_{center}: 0.15$
R_L
ν

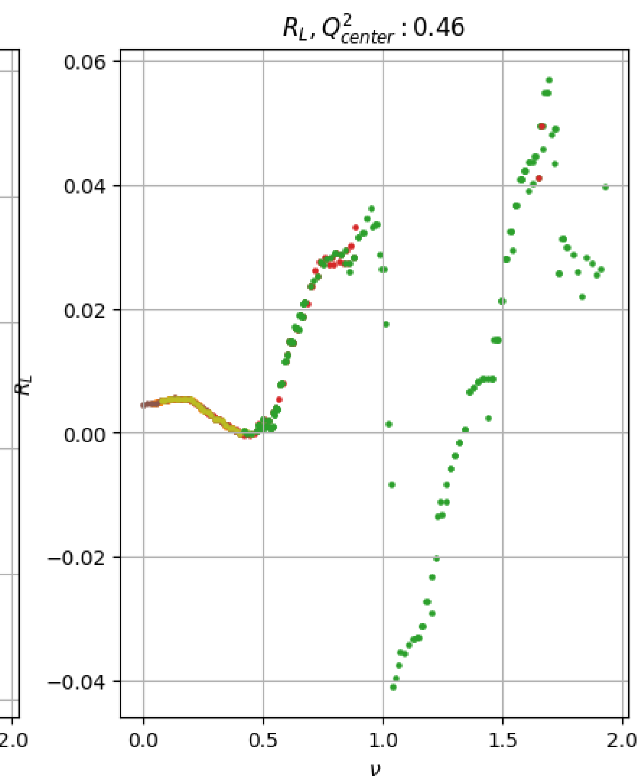


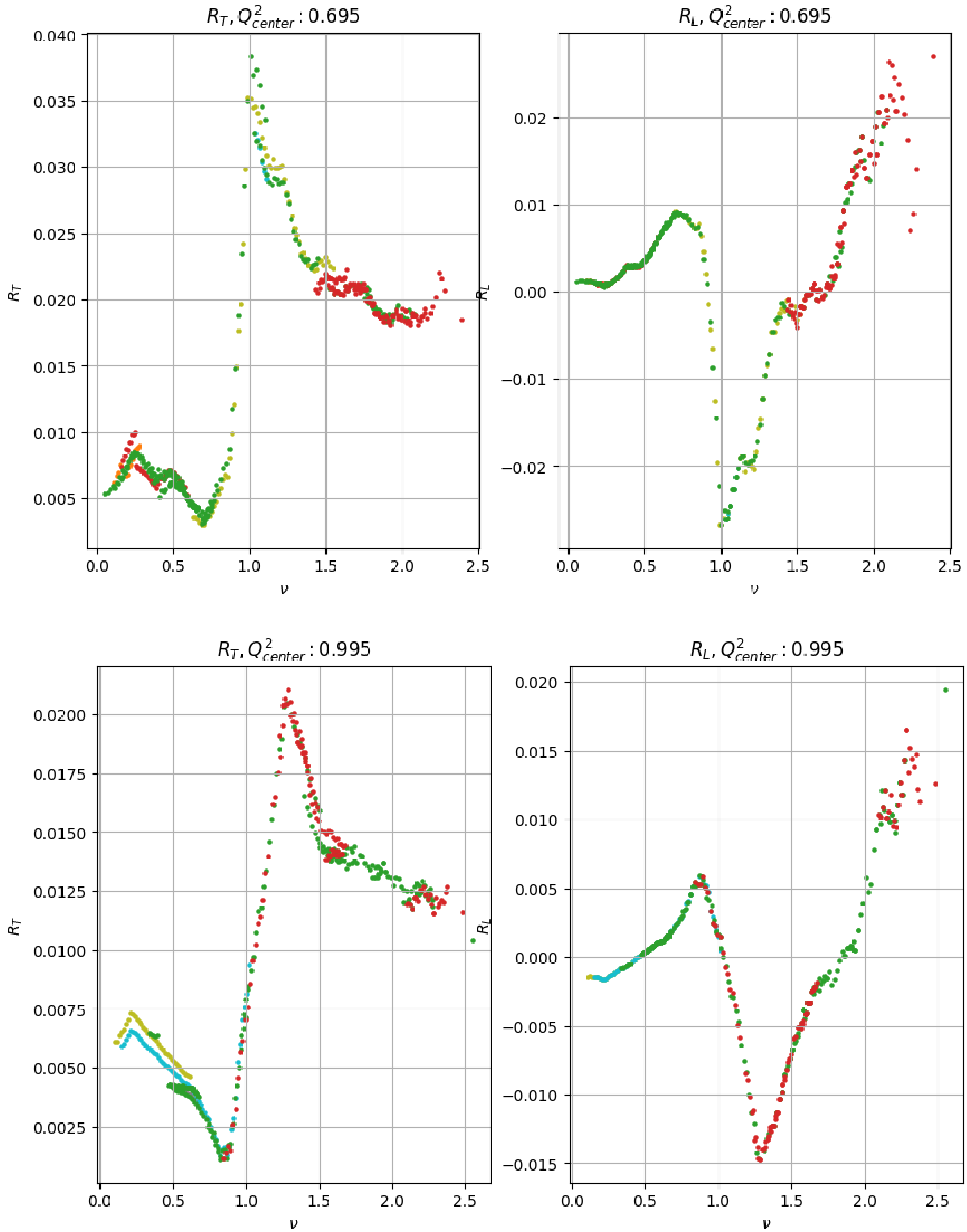
$R_T, Q^2_{center}: 0.215$
$R_L, Q^2_{center}: 0.215$
$R_T, Q^2_{center}: 0.28$
$R_L, Q^2_{center}: 0.28$
R_T
R_L
ν

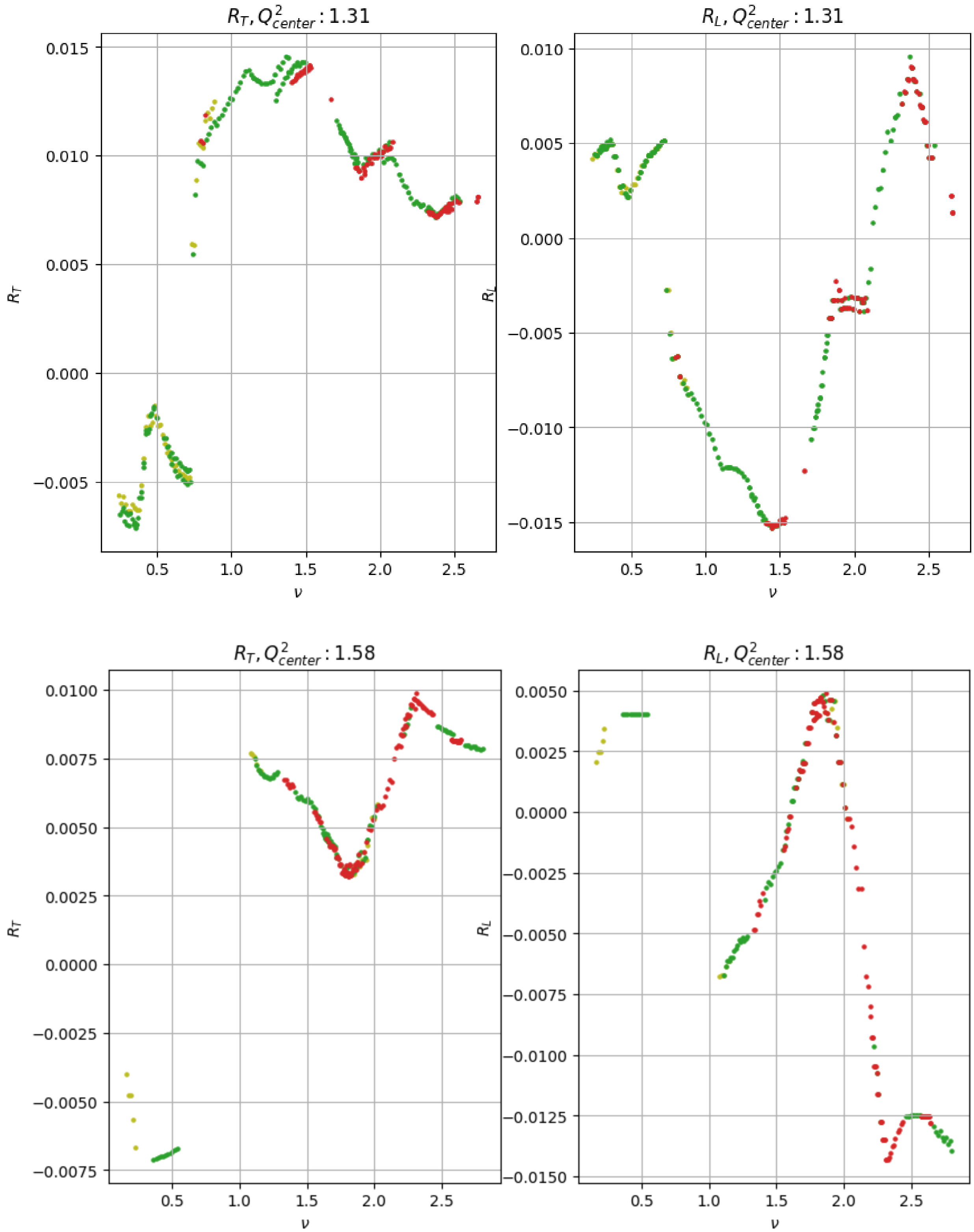

$R_T, Q^2_{center}: 0.335$
R_T
ν


$R_L, Q^2_{center}: 0.335$
R_L
ν

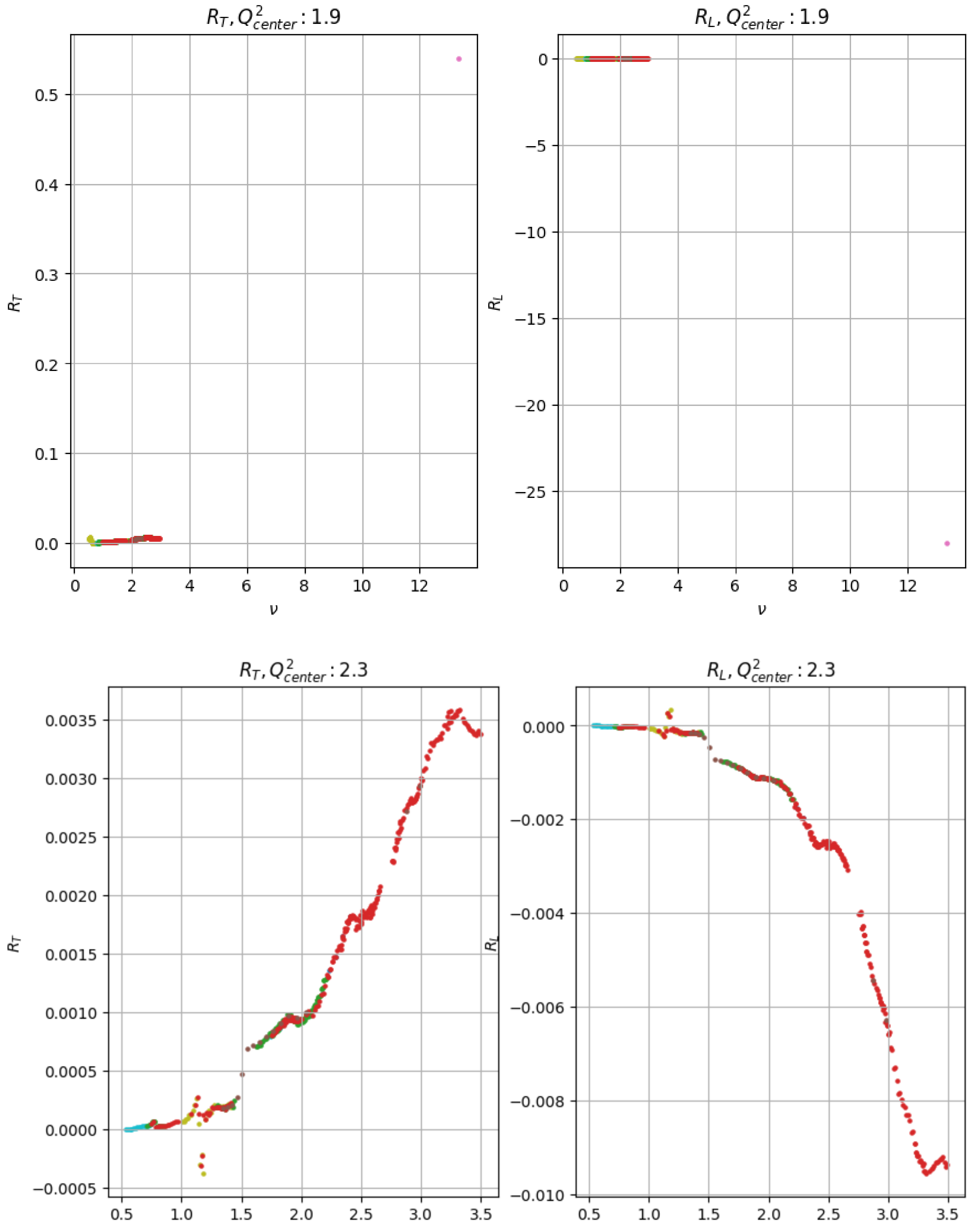

$R_T, Q^2_{center}: 0.46$
R_T
ν


$R_L, Q^2_{center}: 0.46$
R_L
ν


$R_T, Q^2_{center}: 0.695$
$R_L, Q^2_{center}: 0.695$
$R_T, Q^2_{center}: 0.995$
$R_L, Q^2_{center}: 0.995$
R_T
R_L
ν

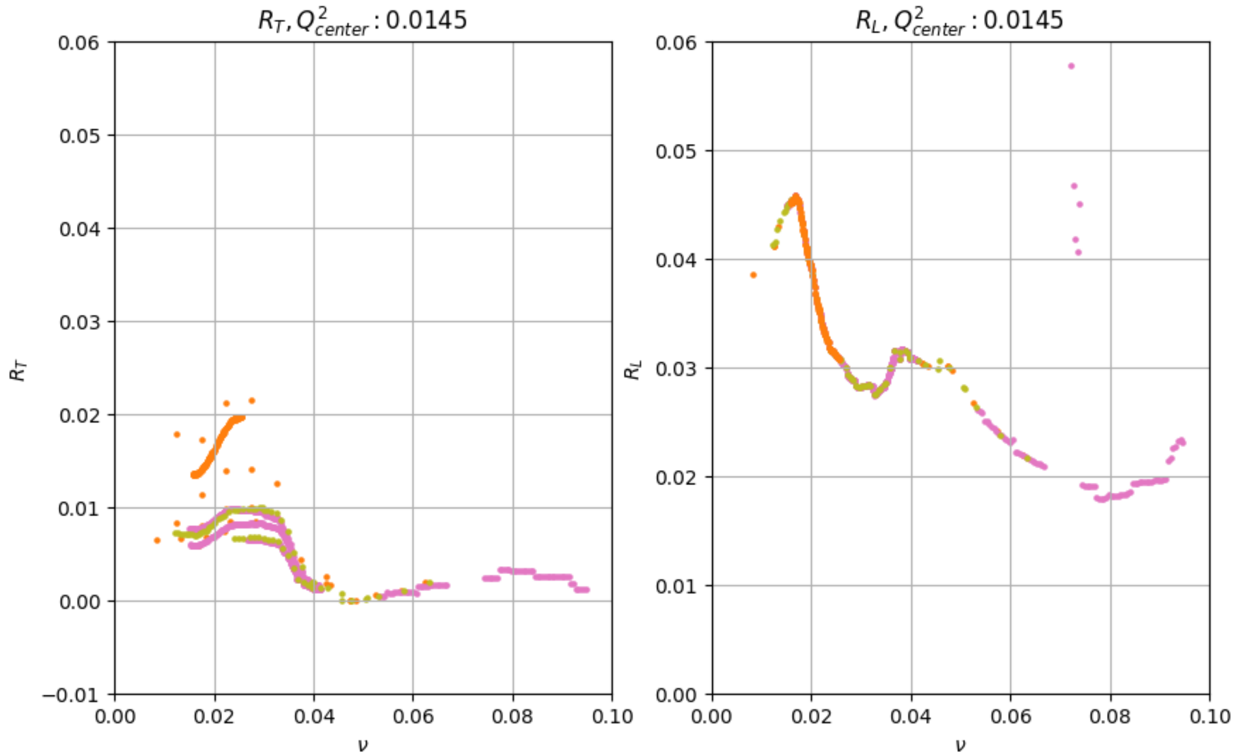


$R_T, Q^2_{center}: 1.31$
$R_L, Q^2_{center}: 1.31$
$R_T, Q^2_{center}: 1.58$
$R_L, Q^2_{center}: 1.58$
R_T
R_L
ν



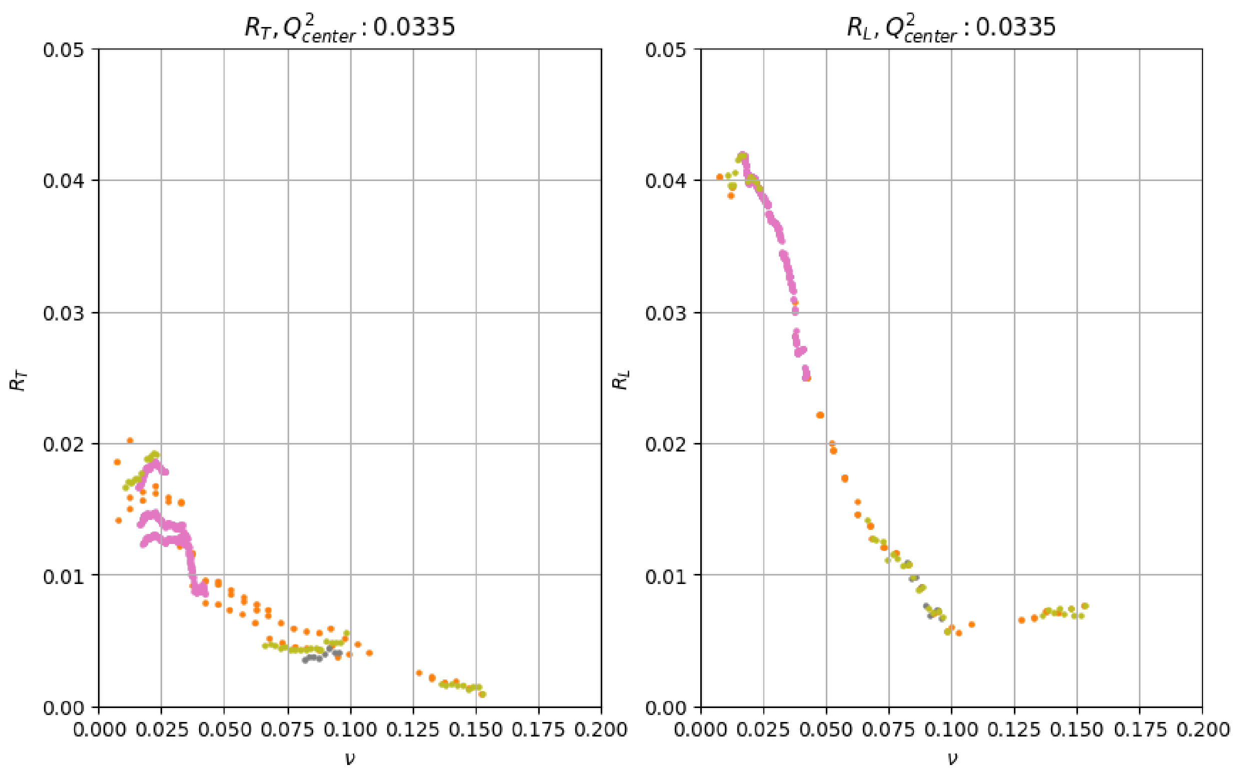
$R_T, Q^2_{center}: 1.9$
$R_L, Q^2_{center}: 1.9$
$R_T, Q^2_{center}: 2.3$
$R_L, Q^2_{center}: 2.3$
R_T
R_L
ν

```
bin_name = Q2bins[0]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.1)
plt.ylim(-0.01, 0.06)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.1)
plt.ylim(0, 0.06)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



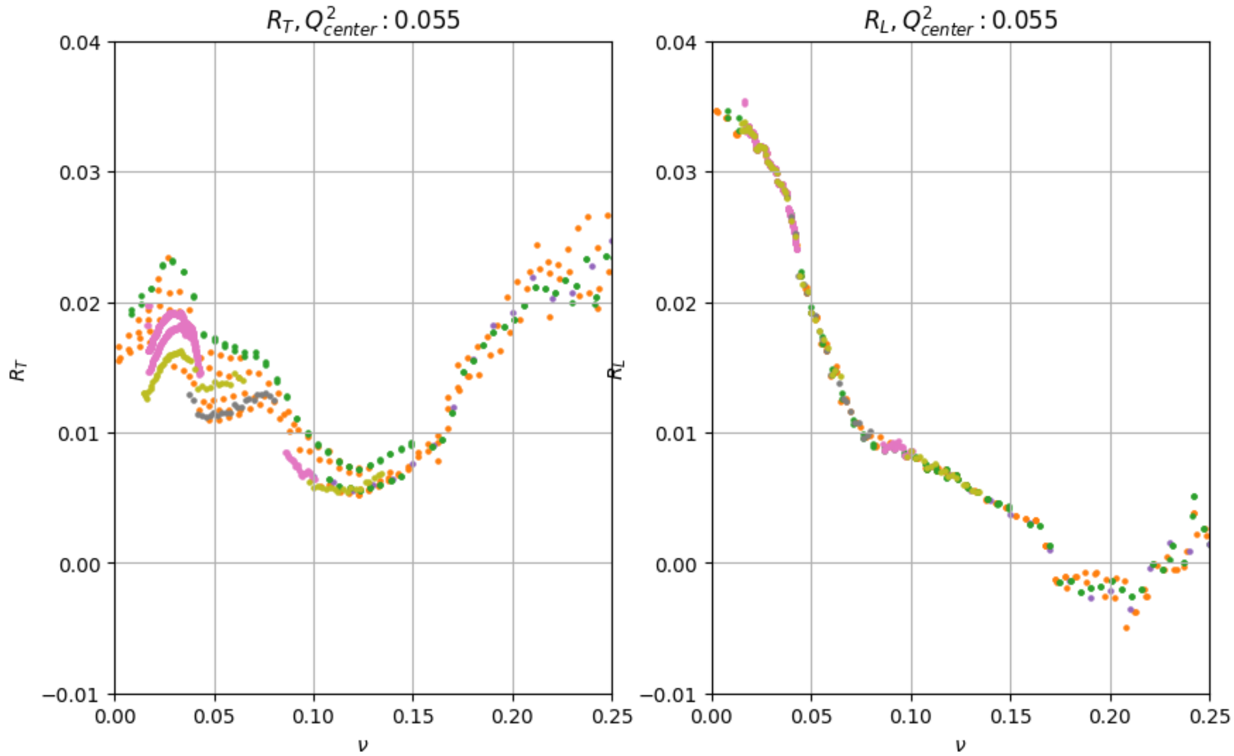

```
bin_name = Q2bins[1]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.2)
plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.2)
plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



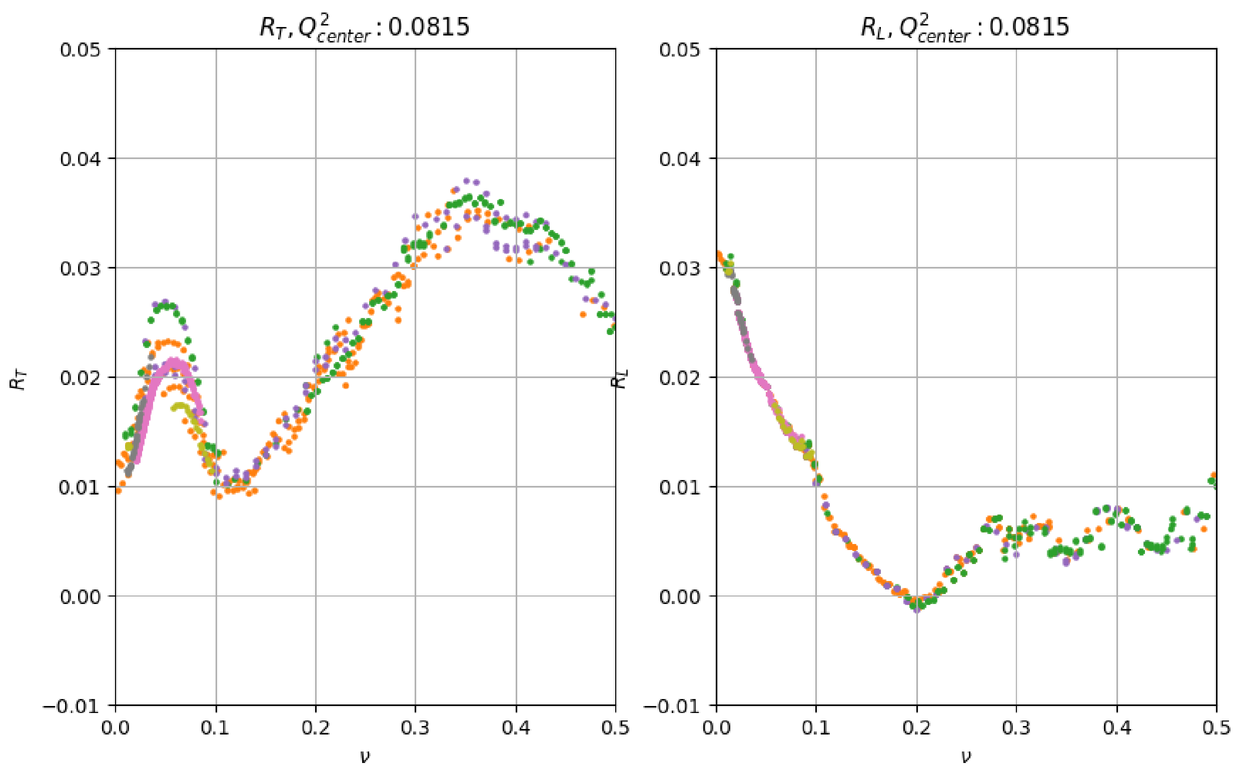

```
bin_name = Q2bins[2]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
```

```
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.25)
plt.ylim(-0.01, 0.04)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.25)
plt.ylim(-0.01, 0.04)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



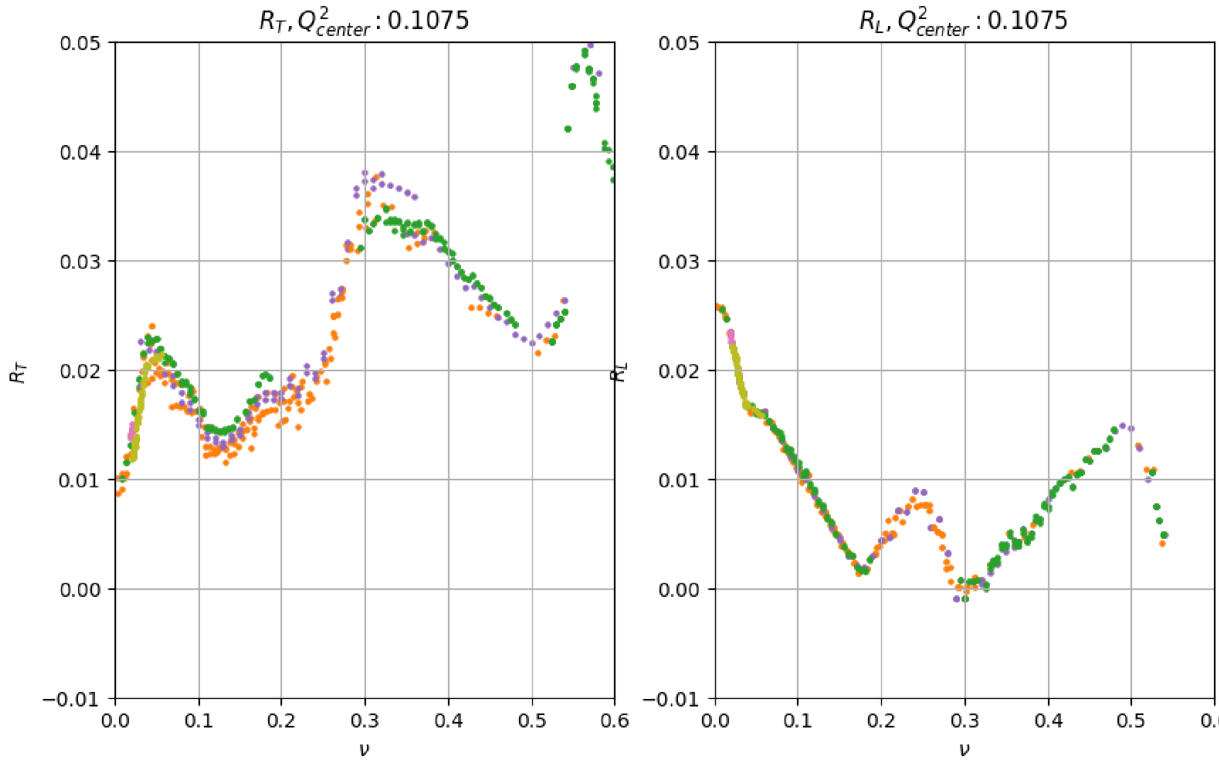

```
bin_name = Q2bins[3]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.5)
plt.ylim(-0.01, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.5)
plt.ylim(-0.01, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



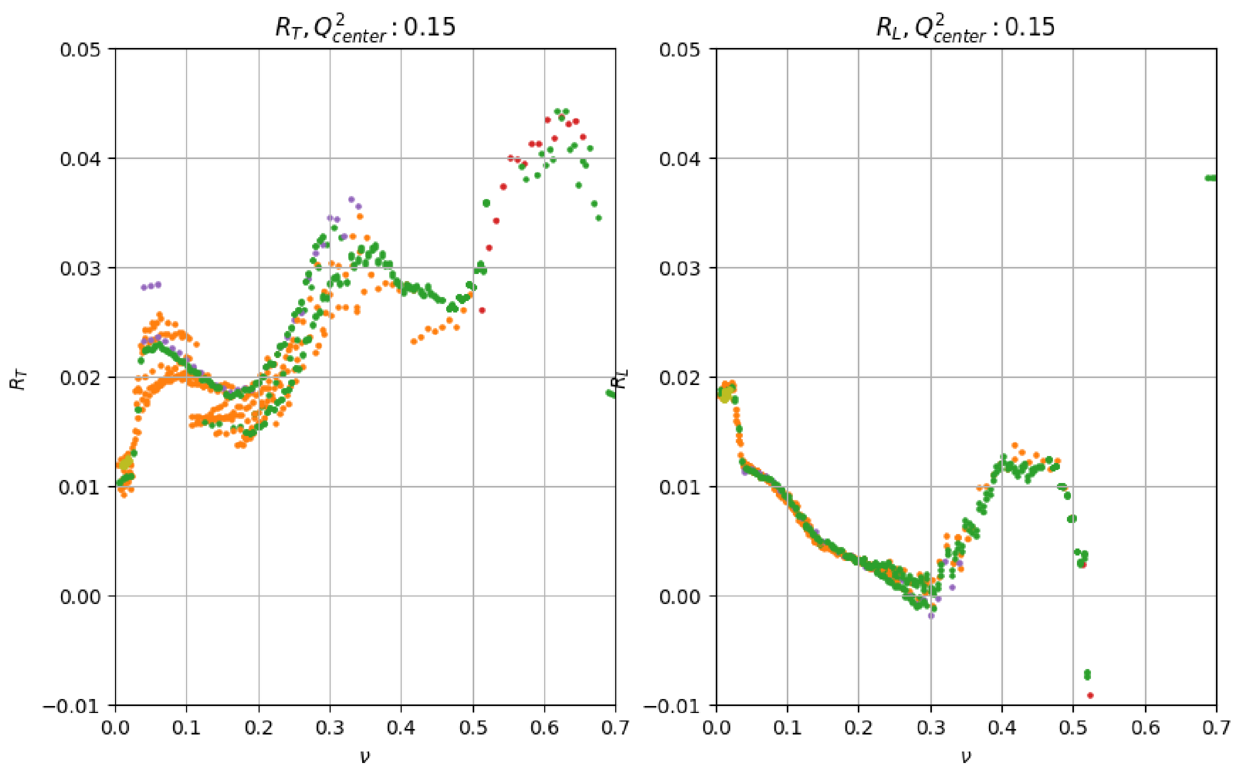

```
bin_name = Q2bins[4]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
```

```
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.6)
plt.ylim(-0.01, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.6)
plt.ylim(-0.01, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



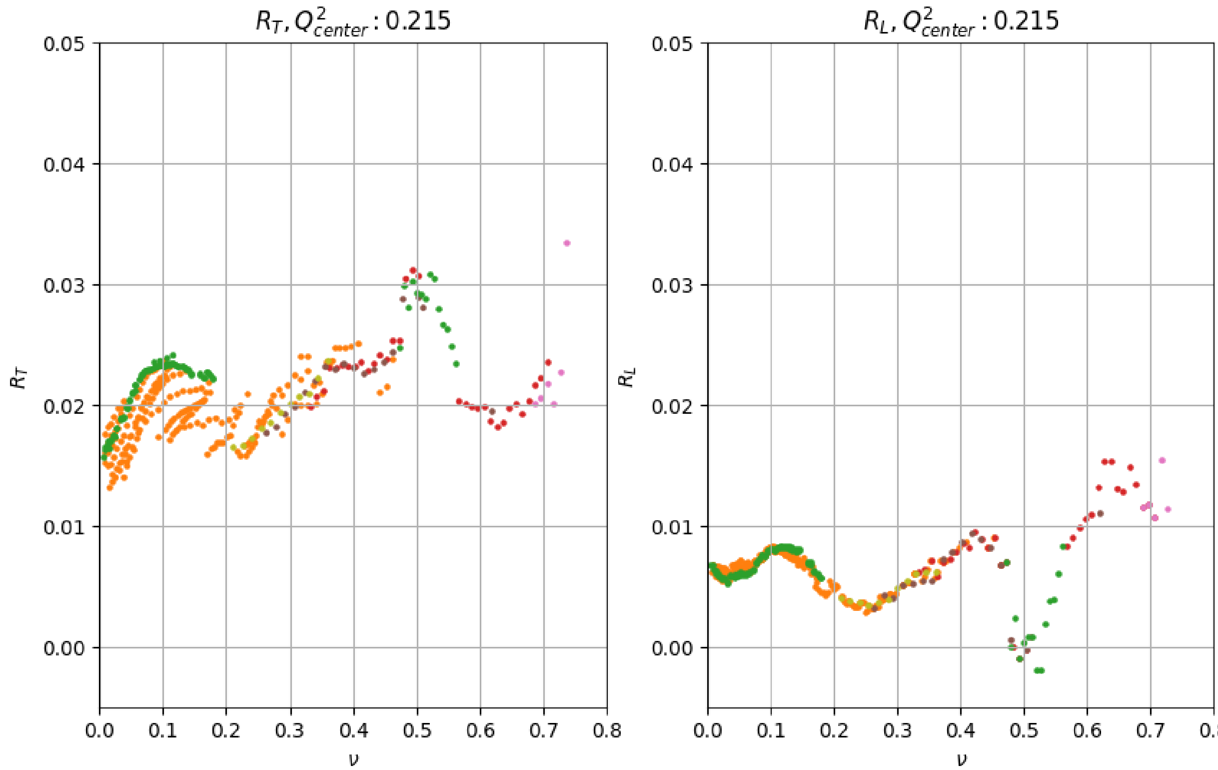

```
bin_name = Q2bins[5]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.7)
plt.ylim(-0.01, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.7)
plt.ylim(-0.01, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



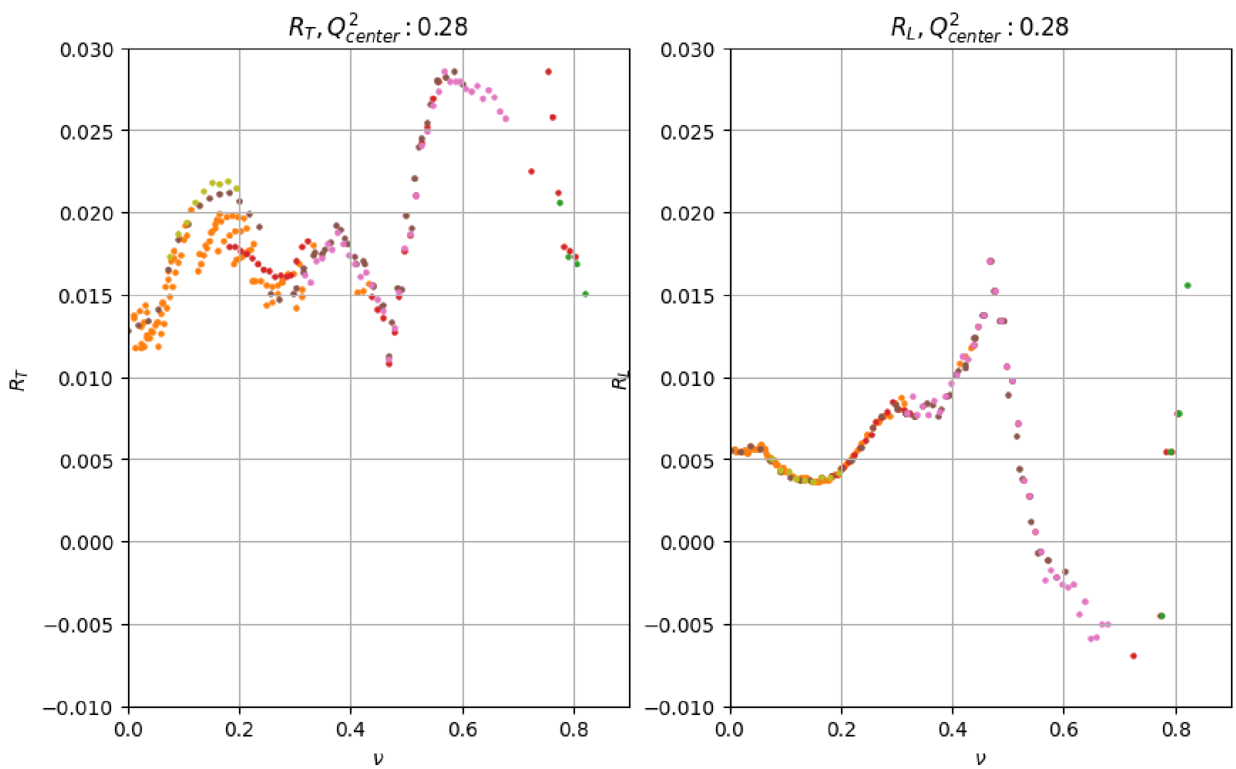

```
bin_name = Q2bins[6]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
```

```
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.8)
plt.ylim(-0.005, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.8)
plt.ylim(-0.005, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



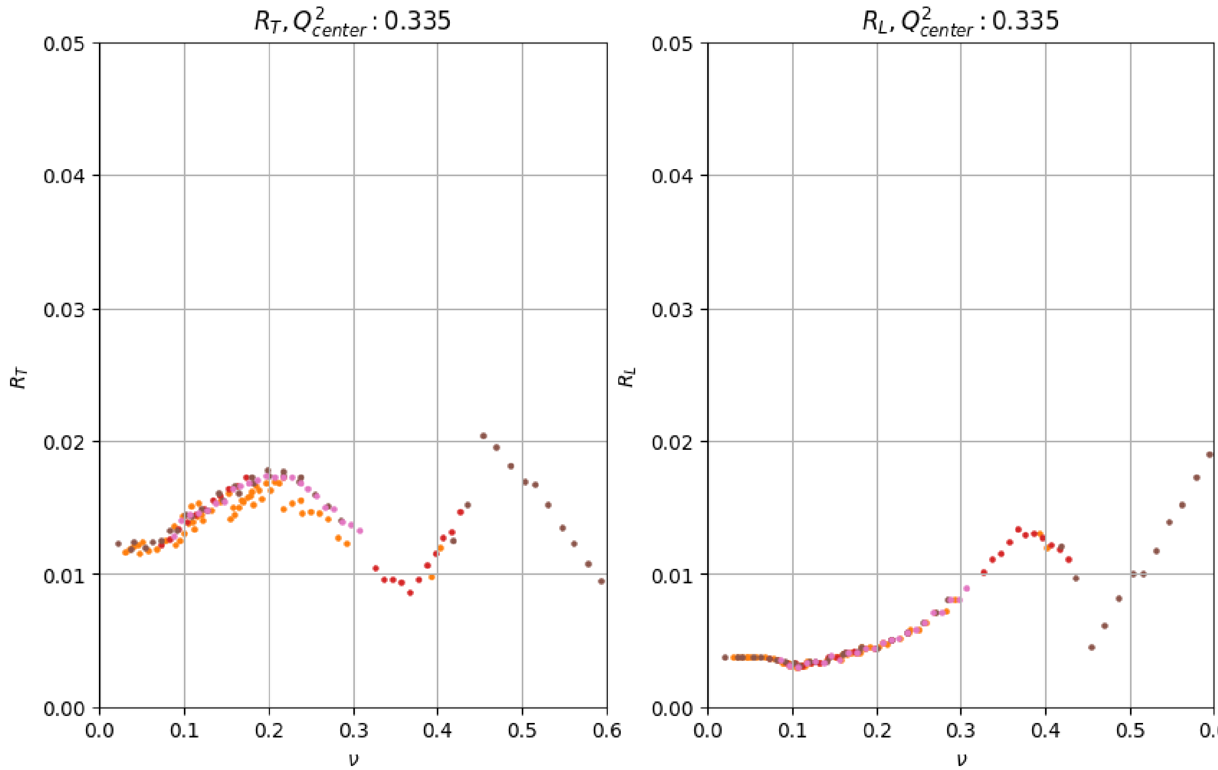

```
bin_name = Q2bins[7]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.9)
plt.ylim(-0.01, 0.03)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.9)
plt.ylim(-0.01, 0.03)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



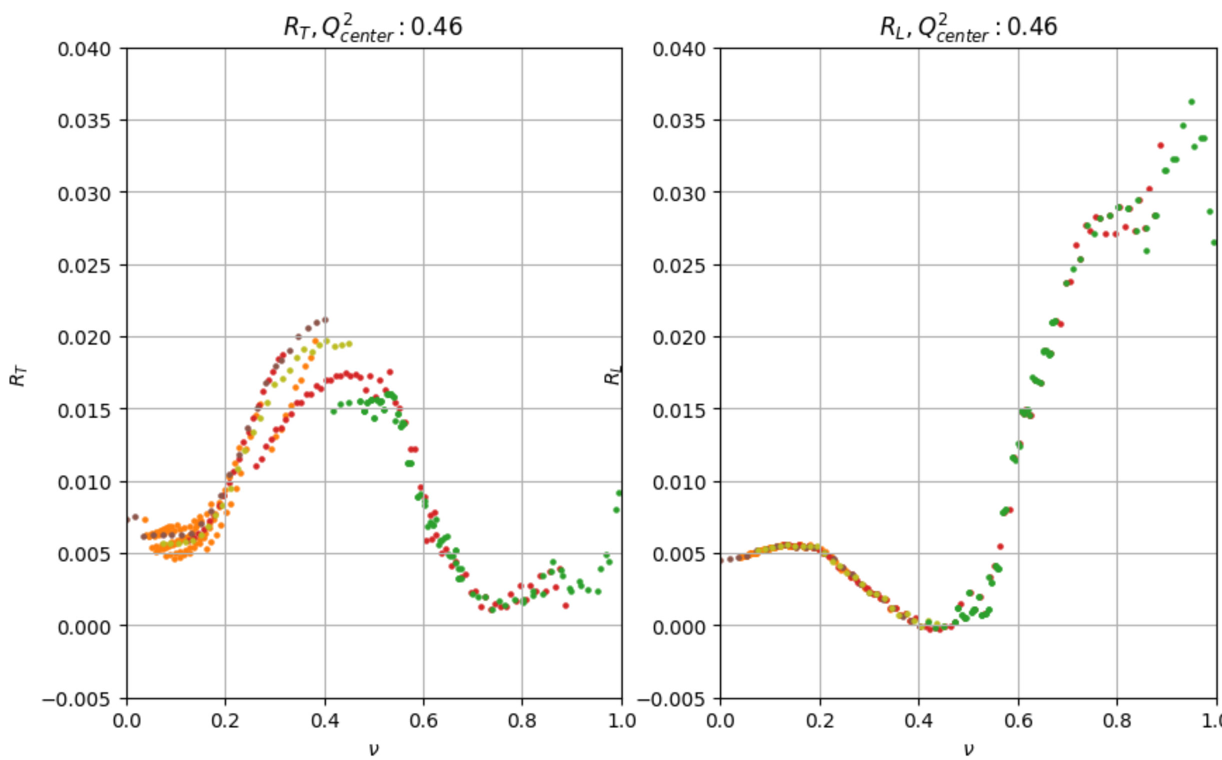

```
bin_name = Q2bins[8]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
```

```
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.6)
plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 0.6)
plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



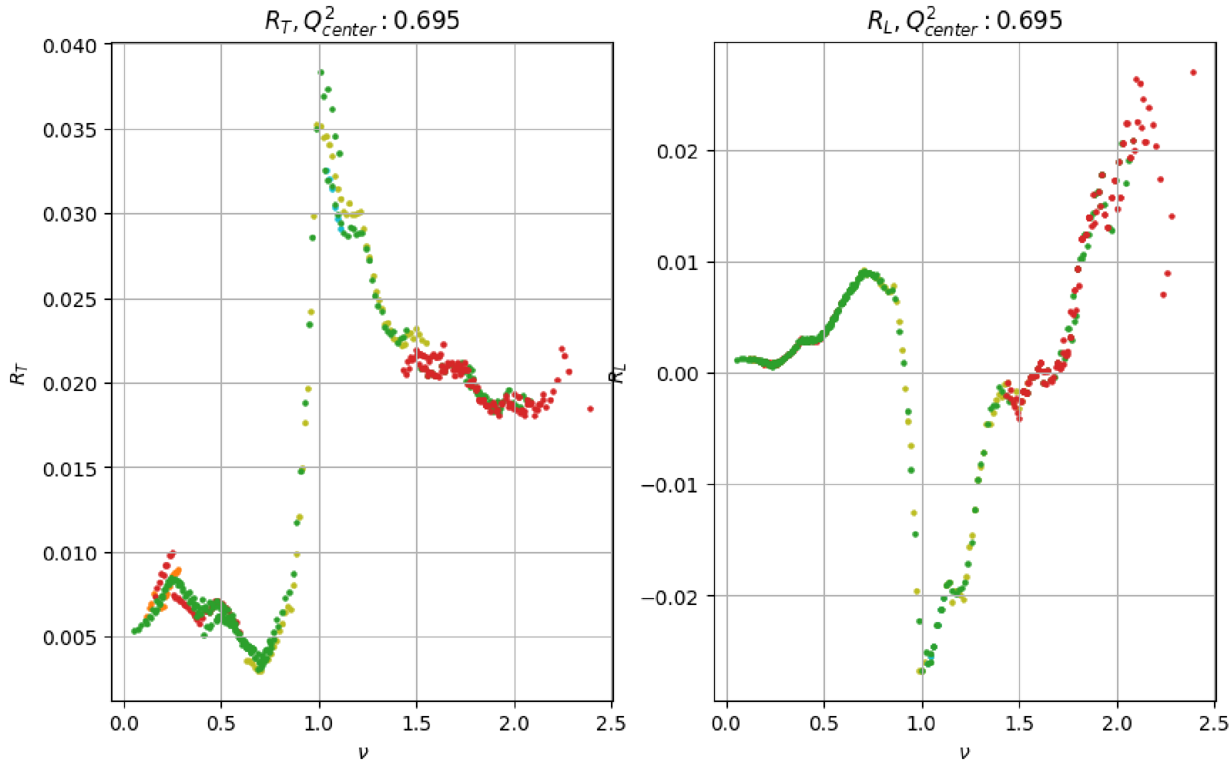

```
bin_name = Q2bins[9]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 1)
plt.ylim(-0.005, 0.04)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 1)
plt.ylim(-0.005, 0.04)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



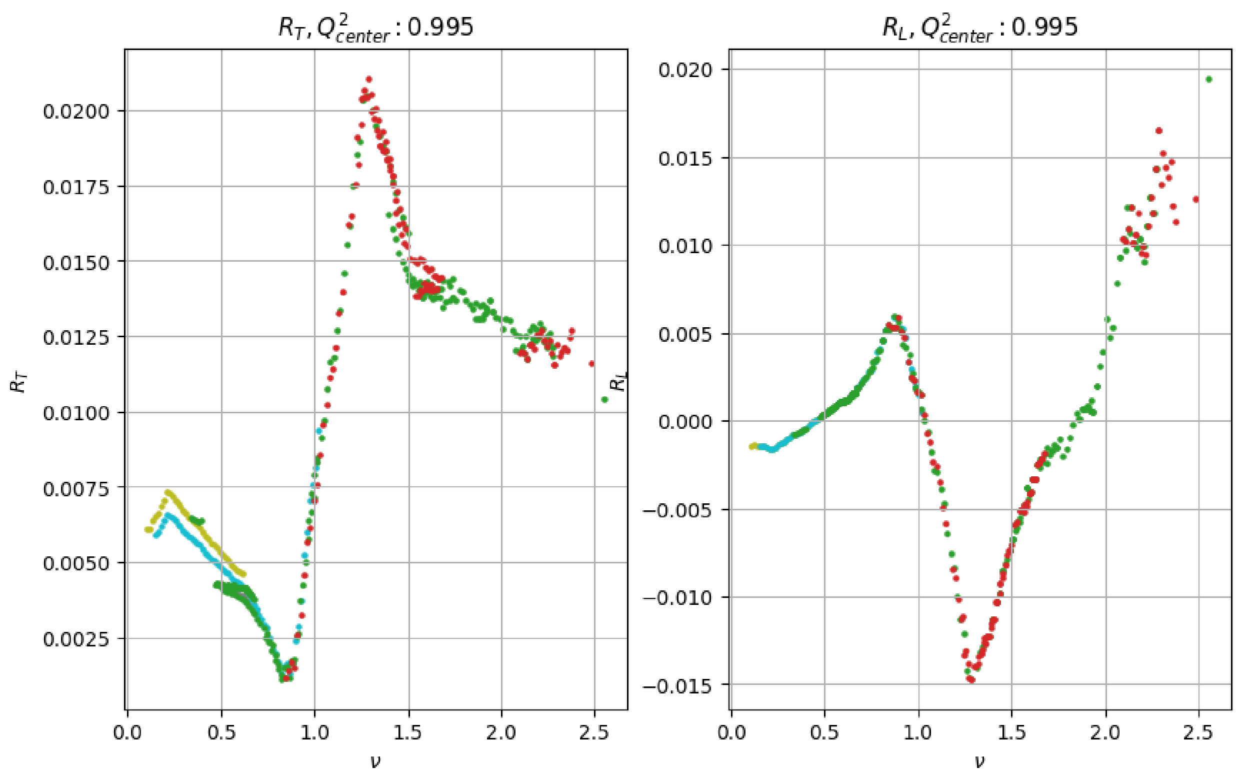

```
bin_name = Q2bins[10]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
```

```
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



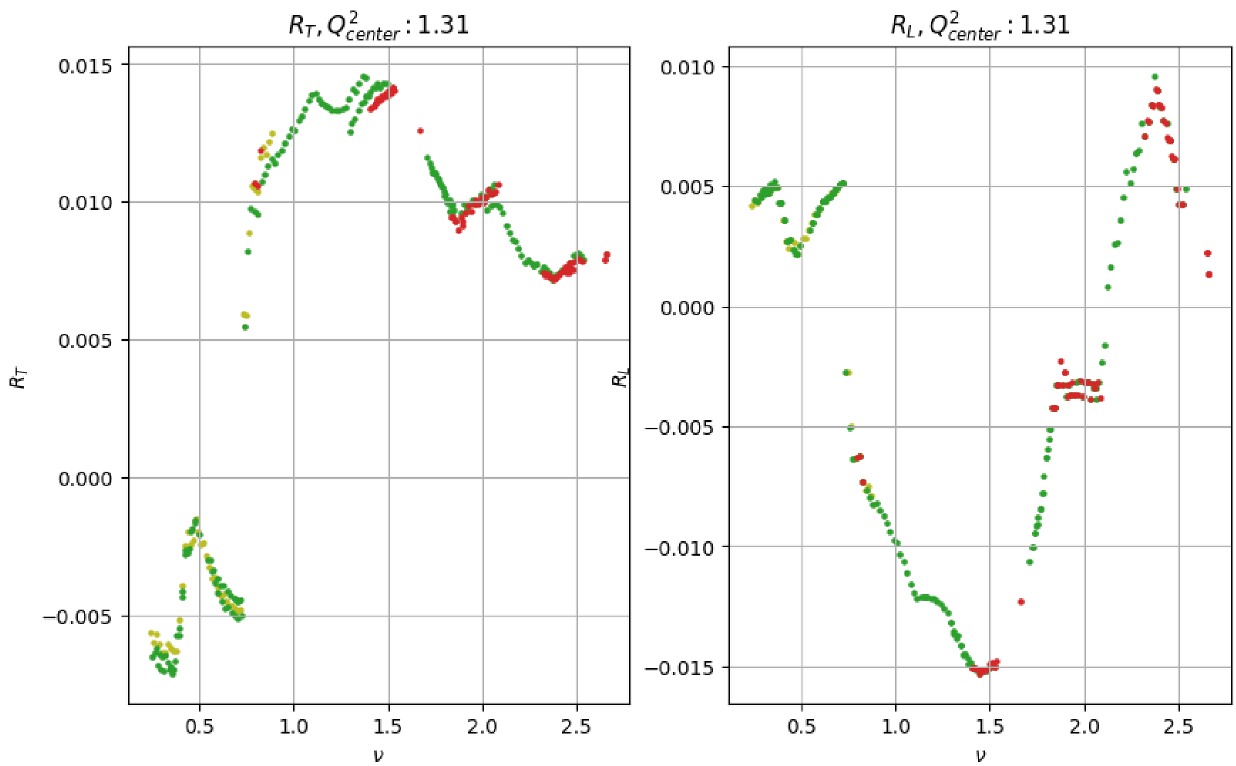

```
bin_name = Q2bins[11]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



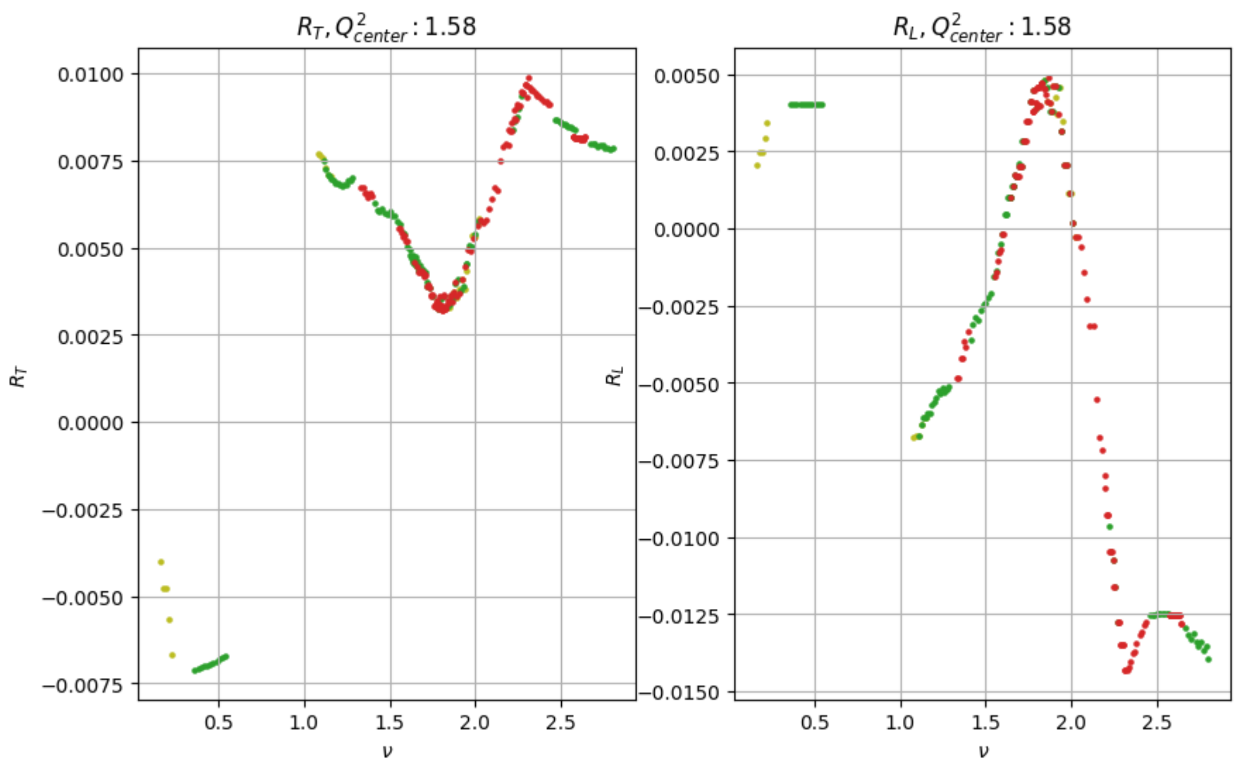

```
bin_name = Q2bins[12]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
```

```
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```



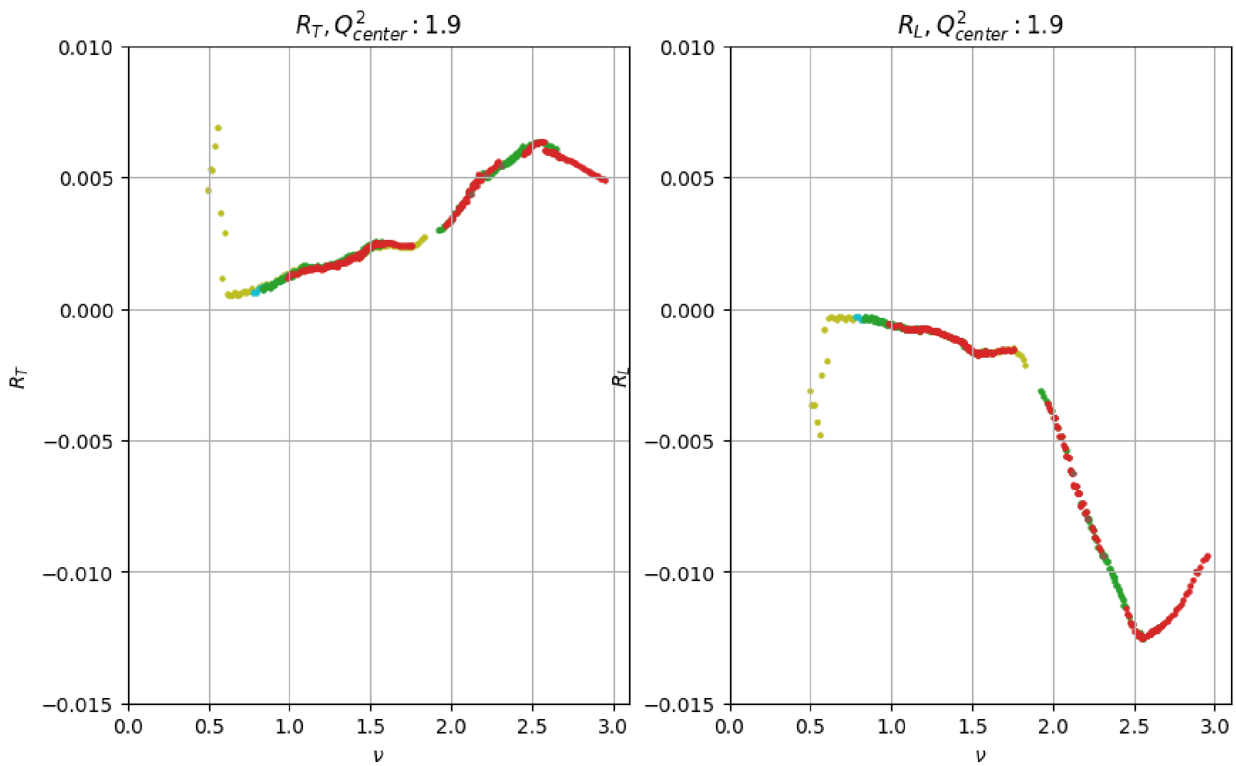

```
bin_name = Q2bins[13]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```




```
bin_name = Q2bins[14]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
```

```
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 3.1)
plt.ylim(-0.015, 0.01)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
plt.xlim(0, 3.1)
plt.ylim(-0.015, 0.01)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```




```
bin_name = Q2bins[15]
fig = plt.figure(figsize=(10, 6))
Q2center = Q2bin_to_Q2center[bin_name]

# bin = df.loc[df["bin"]==bin_name]
bin = df.loc[(df["bin"]==bin_name) & (df["RL"]!=0) & (df["RT"]!=0)]
plt.subplot(1, 2, 1)  # 1 row, 2 columns, and this is the first
subplot
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RT"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_T$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_T,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Plot your data here
# Create the second plot on the right
plt.subplot(1, 2, 2)  # 1 row, 2 columns, and this is the second
subplot
# Plot your data here
```

```
for i in range(0,22):
    plt.scatter((bin.loc[(bin["dataSet"] == i)])
["nu"],bin.loc[(bin["dataSet"] == i)]["RL"],label=i,s=5)
    # plt.scatter(bin["nu"],bin["RT"],s=5)
# plt.xlim(0, 0.6)
# plt.ylim(0, 0.05)
plt.xlabel("$\\nu$")
plt.ylabel("$R_L$")
# plt.title("$Q^2$:"+bin+": data_size:"+str(len(picked))+"
duplicated:"+str(len(dup)))
plt.title("$R_L,Q^2_{center}:$"+str(Q2center))
plt.grid()

# Adjust spacing between subplots
# plt.tight_layout()
# plt.legend()

# Display the plots
plt.show()
```