WP2outline

March 16, 2018

In [1]:
```
from plotting import *
from CVZ import *
%matplotlib inline
# get simulation
self = loadpickle('pickles/RVInformationGP_Nrvge10')
#self = loadpickle('pickles/RVInformationGP')
```

In [2]:
```
# write function that returns indices of planets of interest based on their median obser
def get_planets(tobs, g, N=0, sort=True):
    tobsinds = np.argsort(tobs[g]) if sort else np.arange(tobs[g].size)
    inds = np.arange(tobs.size)[g][tobsinds]
    return inds if N == 0 else inds[:int(N)]
```

In [3]:
```
# get K2 targets
fname = 'input_data/K2planets_Kdwarfs.csv'
starname, planetname = np.genfromtxt(fname,delimiter=',',skip_header=75,usecols=(1,2),dt
starnames = np.array(['%s%s'%(starname[i], planetname[i]) for i in range(starname.size)]
PK2,aK2,rpK2,TeffK2,MsK2,RsK2,JK2 = np.genfromtxt(fname,delimiter=',',skip_header=75,use
assert PK2.size == starnames.size
aK2[np.isnan(aK2)] = rvs.semimajoraxis(PK2, MsK2, 0)[np.isnan(aK2)]
rpK2 *= 11.21
TpK2 = TeffK2 * np.sqrt(rvs.Rsun2m(RsK2)/(2*rvs.AU2m(aK2)))
muK2 = np.repeat(2, TpK2.size)
muK2[rpK2 <= 2] = 30.
mpK2 = rvs.kg2Mearth(9.8*rvs.Rearth2m(rpK2)**2 / 6.67e-11)
transmissionK2_ppm = rvs.transmission_spectroscopy_depth(RsK2, mpK2, rpK2, TpK2, muK2)
```

0.1 Measuring the 3σ mass of one temperate Earth-sized planet

In [4]:
```
ind = 1451
inds = np.array([ind])
print 'Orbital period = %.3f days'%self.Ps_med[ind]
print 'Planet radius = %.3f Earth radii'%self.rps_med[ind]
print 'Planet mass = %.3f Earth masses'%self.mps_med[ind]
print 'J = %.3f'%self.Jmags_med[ind]
print 'Stellar effective temperature = %i K'%self.Teffs_med[ind]
```

```
Orbital period = 26.300 days
Planet radius = 1.292 Earth radii
```

```
Planet mass = 2.660 Earth masses
J = 10.270
Stellar effective temperature = 3284 K
```

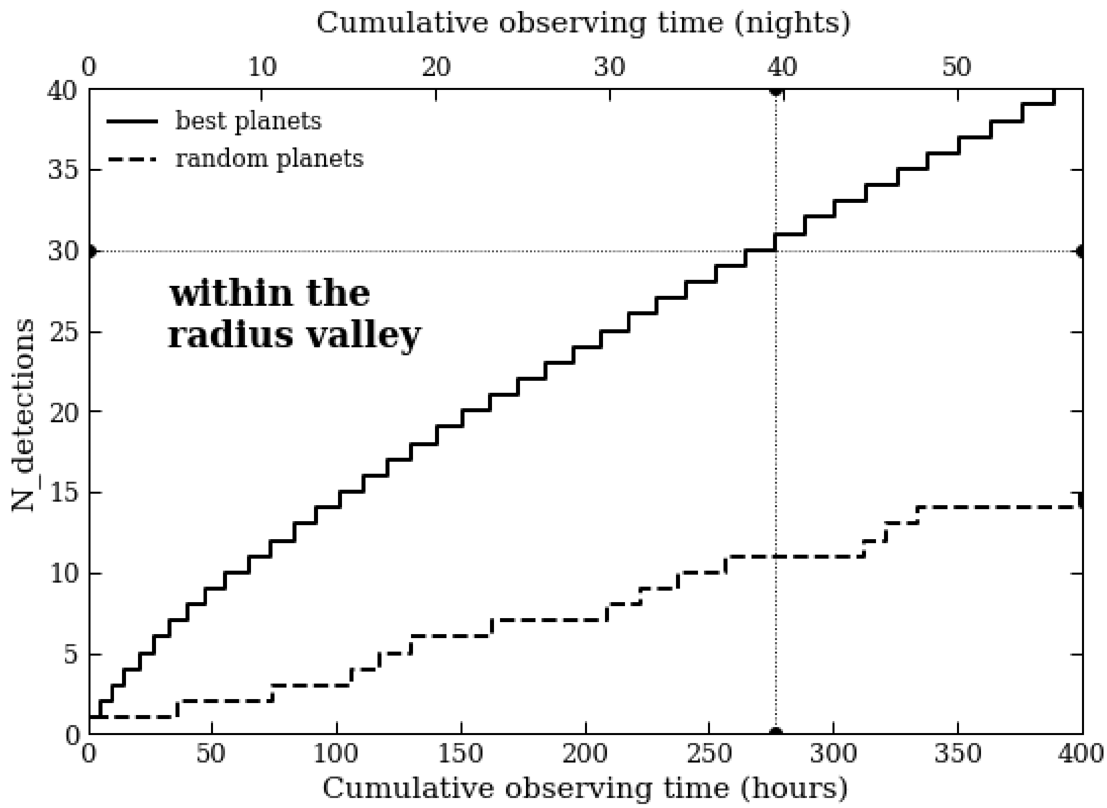
In [5]:
```
print 'Exposure time = %.2f minutes'%self.texps_med_N[ind]
print 'RV precision = %.2f m/s'%self.sigmaRV_phot_med_N[ind]
print 'Median effective RV rms = %.2f m/s'%self.sigmaRV_eff_med_N[ind]
print 'Median number of RV measurements required = %.1f'%self.NrvGPs_med_N[ind]
print 'Total observing time = %.1f hours (i.e. %.1f nights)'%(self.tobsGPs_med_N[ind], s
tobs_WP2 = self.tobsGPs_med_N[ind]
```

```
Exposure time = 10.00 minutes
RV precision = 2.98 m/s
Median effective RV rms = 7.86 m/s
Median number of RV measurements required = 248.8
Total observing time = 41.5 hours (i.e. 5.9 nights)
```

0.2 Measuring the 5σ mass of the 30 'best' planets within the radius valley

In [6]:
```
Nf1 = 30
scale = (.327/.189)**2  # 3 -> 5 sigma
g = (self.rps_med >= 1.5) & (self.rps_med <= 2.5) & (self.decs_med > -15)
tobs = np.append(0, np.cumsum(np.sort(self.tobsGPs_med_N[g]*scale)))
tobs2 = np.append(0, np.cumsum(self.tobsGPs_med_N[g]*scale))
Ndet= np.arange(tobs.size)
fig = plt.figure(figsize=(9,6))
ax1 = fig.add_subplot(111)
ax1.plot(tobs, Ndet, 'k-', lw=2, drawstyle='steps', label='best planets')
ax1.plot(tobs2, Ndet, 'k--', lw=2, drawstyle='steps', label='random planets')
ax1.axhline(Nf1, ls=':', lw=.8), ax1.axvline(tobs[Ndet==Nf1], ls=':', lw=.9)
tobs_WP2 = np.append(tobs_WP2, tobs[Ndet==Nf1])
ax1.set_xlim((0,4e2)), ax1.set_ylim((0,40)), ax1.legend(loc='upper left', fontsize=12)
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detections')
ax1.text(.08, .6, 'within the\nradius valley', transform=ax1.transAxes, fontsize=18, wei
ax2 = ax1.twiny()
ax2.set_xlim((0,4e2/7)), ax2.set_xlabel('Cumulative observing time (nights)', labelpad=1
```

Out[6]:
```
((0, 57.142857142857146), <matplotlib.text.Text at 0x1a14a9b6d0>)
```

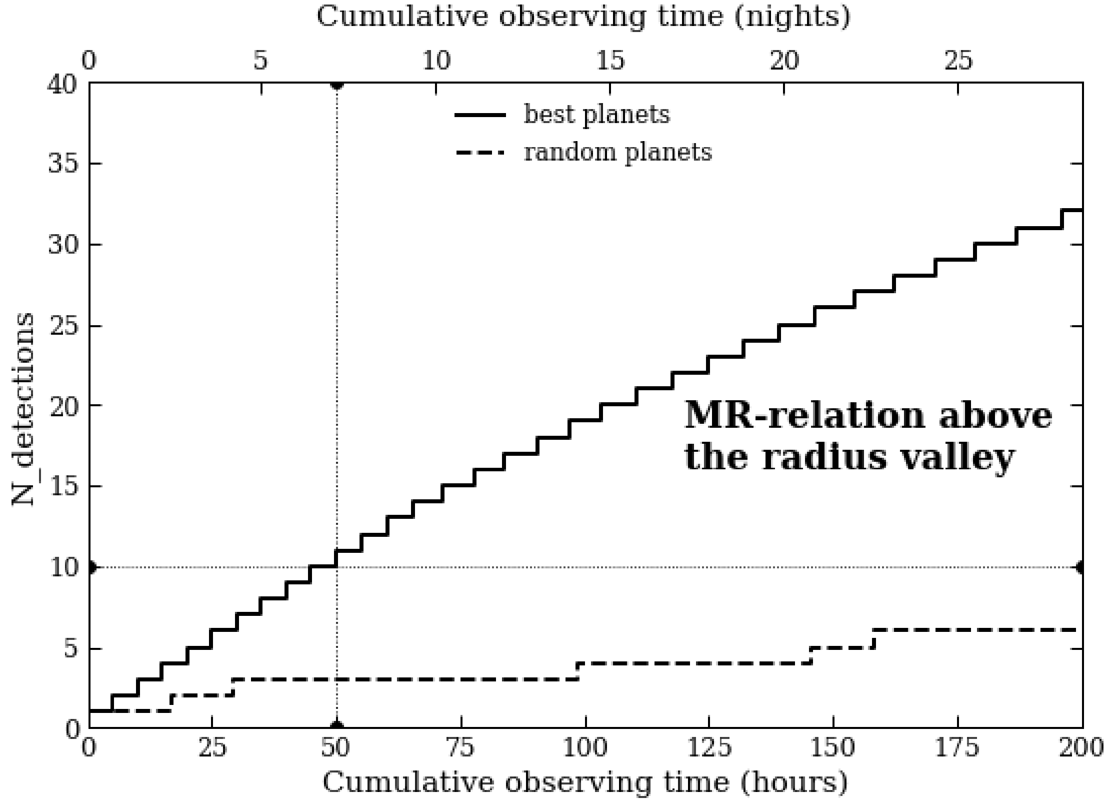



```
In [7]: # save TOI indices
        inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf1))
```

0.3 Extending the MR relation with 10 planets with $r_p \in [2.5, 4] R_\oplus$

```
In [8]: Nf2 = 10
        g = (self.rps_med > 2.5) & (self.rps_med <= 4) & (self.decs_med > -15)
        tobs = np.append(0, np.cumsum(np.sort(self.tobsGPs_med_N[g]*scale)))
        tobs2 = np.append(0, np.cumsum(self.tobsGPs_med_N[g]*scale))
        Ndet= np.arange(tobs.size)
        fig = plt.figure(figsize=(9,6))
        ax1 = fig.add_subplot(111)
        ax1.plot(tobs, Ndet, 'k-', lw=2, drawstyle='steps', label='best planets')
        ax1.plot(tobs2, Ndet, 'k--', lw=2, drawstyle='steps', label='random planets')
        ax1.axhline(Nf2, ls=':', lw=.8), ax1.axvline(tobs[Ndet==Nf2], ls=':', lw=.9)
        tobs_WP2 = np.append(tobs_WP2, tobs[Ndet==Nf2])
        ax1.set_xlim((0,2e2)), ax1.set_ylim((0,40)), ax1.legend(loc='upper center', fontsize=12)
        ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detections')
        ax1.text(.6, .4, 'MR-relation above\nthe radius valley', transform=ax1.transAxes, fontsi
        ax2 = ax1.twiny()
        ax2.set_xlim((0,2e2/7)), ax2.set_xlabel('Cumulative observing time (nights)', labelpad=1
```

```
Out[8]: ((0, 28.571428571428573), <matplotlib.text.Text at 0x1a166c75d0>)
```

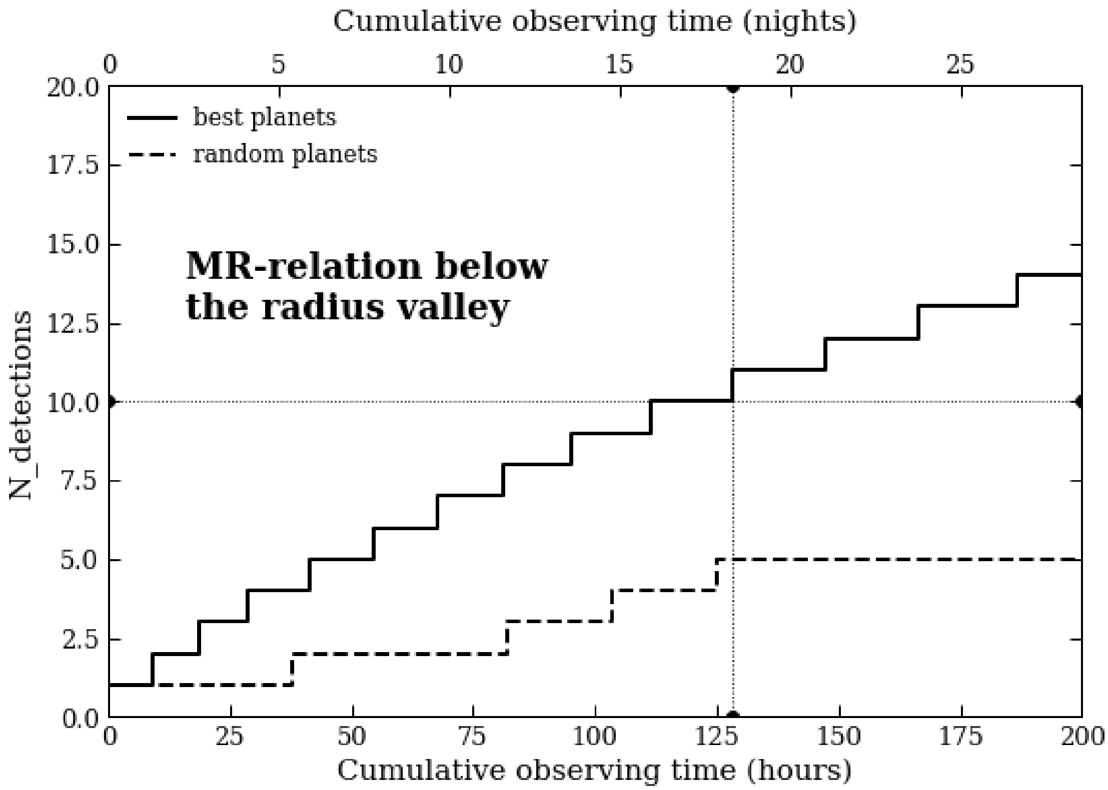



```
In [9]: inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf2))
```

0.4 Extending the MR relation with 10 more planets with $r_p < 1.5R_\oplus$

```
In [10]: Nf3=10
         g = (self.rps_med < 1.5) & (self.decs_med > -15)
         tobs = np.append(0, np.cumsum(np.sort(self.tobsGPs_med_N[g]*scale)))
         tobs2 = np.append(0, np.cumsum(self.tobsGPs_med_N[g]*scale))
         Ndet= np.arange(tobs.size)
         fig = plt.figure(figsize=(9,6))
         ax1 = fig.add_subplot(111)
         ax1.plot(tobs, Ndet, 'k-', lw=2, drawstyle='steps', label='best planets')
         ax1.plot(tobs2, Ndet, 'k--', lw=2, drawstyle='steps', label='random planets')
         ax1.axhline(Nf3, ls=':', lw=.8), ax1.axvline(tobs[Ndet==Nf3], ls=':', lw=.9)
         tobs_WP2 = np.append(tobs_WP2, tobs[Ndet==Nf3])
         ax1.set_xlim((0,2e2)), ax1.set_ylim((0,20)), ax1.legend(loc='upper left', fontsize=12)
         ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detections')
         ax1.text(.08, .63, 'MR-relation below\nthe radius valley', transform=ax1.transAxes, fon
         ax2 = ax1.twiny()
         ax2.set_xlim((0,2e2/7)), ax2.set_xlabel('Cumulative observing time (nights)', labelpad=
```

```
Out[10]: ((0, 28.571428571428573), <matplotlib.text.Text at 0x1a19a434d0>)
```




```
In [11]: inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf3))
```

0.5 Measuring 5σ masses of TOIs amenable to transmission spectroscopy

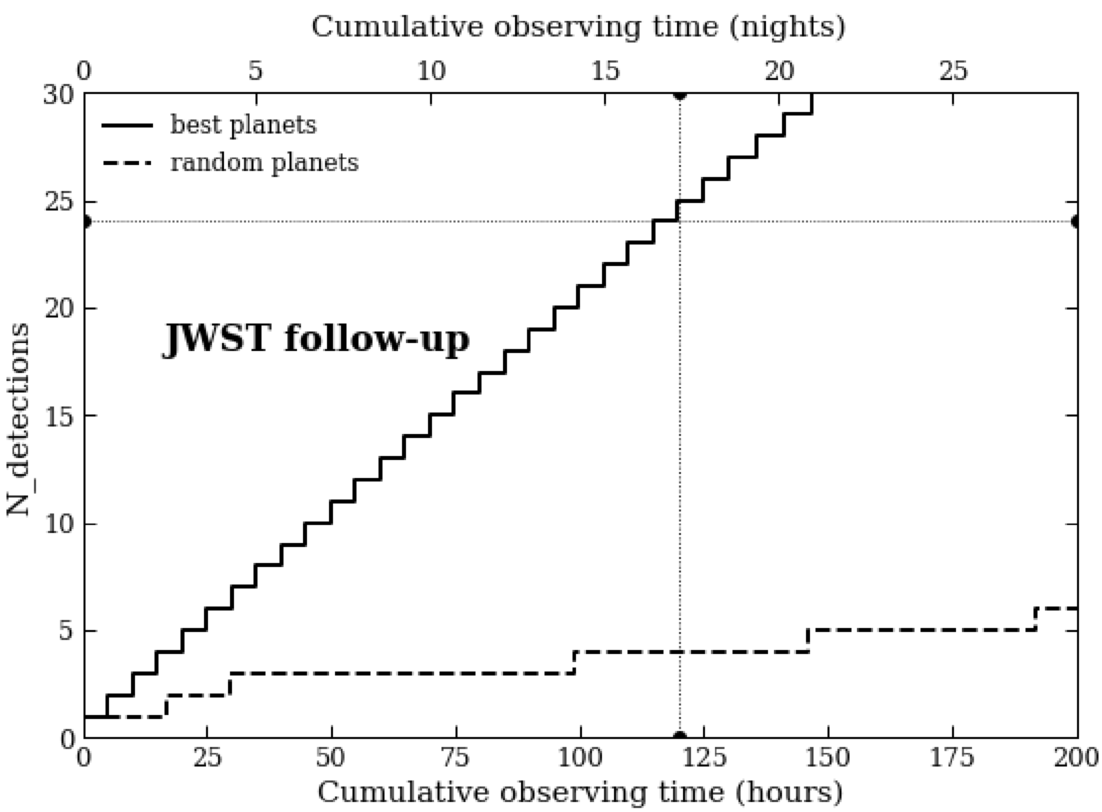
```
In [12]: # set number of WP2 nights available for RV follow-up
         transmission_spec_nights = 5
         trappist1_nights = 7
         total_nights = 100 - transmission_spec_nights - trappist1_nights
         tobs_remaining = total_nights*7. - tobs_WP2.sum()
```

```
In [13]: # get transmission signals
         Tps = self.Teffs_med * np.sqrt(rvs.Rsun2m(self.Rss_med)/(2*rvs.AU2m(rvs.semimajoraxis(s
         mus = np.repeat(2, Tps.size)
         mus[self.rps_med <= 2] = 30.
         transmission_ppm = rvs.transmission_spectroscopy_depth(self.Rss_med, self.mps_med, self
```

```
In [14]: gjwst = (np.in1d(np.arange(self.nstars), inds, invert=True)) & (self.Jmags_med > 6) & (
         tobs = np.append(0, np.cumsum(np.sort(self.tobsGPs_med_N[gjwst]*scale)))
         tobs2 = np.append(0, np.cumsum(self.tobsGPs_med_N[gjwst]*scale))
         Ndet= np.arange(tobs.size)
```

```
fig = plt.figure(figsize=(9,6))
ax1 = fig.add_subplot(111)
ax1.plot(tobs, Ndet, 'k-', lw=2, drawstyle='steps', label='best planets')
ax1.plot(tobs2, Ndet, 'k--', lw=2, drawstyle='steps', label='random planets')
Njwst = int(Ndet[abs(tobs-tobs_remaining) == np.min(abs(tobs-tobs_remaining))])
tobs_WP2 = np.append(tobs_WP2, tobs_remaining)
ax1.axhline(Njwst, ls=':', lw=.8), ax1.axvline(tobs_remaining, ls=':', lw=.9)
ax1.set_xlim((0,np.ceil(tobs_remaining/1e2)*1e2)), ax1.set_ylim((0,np.ceil(Njwst/1e1)*1
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detections')
ax1.text(.08, .6, 'JWST follow-up', transform=ax1.transAxes, fontsize=18, weight='semib
ax2 = ax1.twiny()
ax2.set_xlim((0,np.ceil(tobs_remaining/1e2)*1e2/7)), ax2.set_xlabel('Cumulative observi
```

Out[14]: ((0, 28.571428571428573), <matplotlib.text.Text at 0x1a152b7f50>)




```
In [15]: inds = np.append(inds, get_planets(self.tobsGPs_med_N, gjwst, Njwst))
         print 'We detect %i random potential JWST targets in %.1f hours (i.e. %.1f nights)'%(Nj
```

We detect 24 random potential JWST targets in 120.1 hours (i.e. 17.2 nights)

0.6 Summary of WP2 time allocations and planet populations

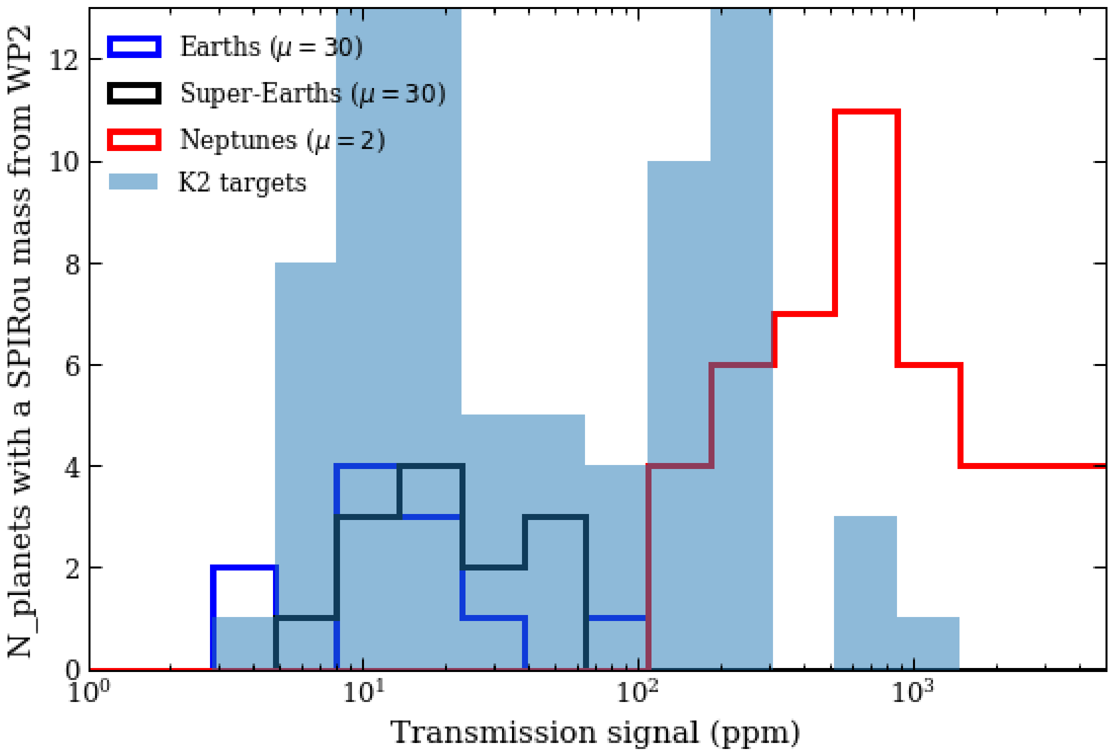
```
In [16]: labels = ['1 temperature Earth-sized planet','%i planets within the radius valley'%Nf1,
         for i in range(tobs_WP2.size):
             print 'Measuring %s requires %.1f hours (i.e. %.1f nights).'%(labels[i], tobs_WP2[i
         print 'Total observing time for %i TESS targets = %.1f hours (i.e. %.1f nights)'%(inds.
         print '\nTransmission spectroscopy is allocated %.1f hours (i.e. %.1f nights)'%(transmi
         print 'Monitoring of the TRAPPIST-1 system is allocated %.1f hours (i.e. %.1f nights)'%
         tot_time = tobs_WP2.sum() + (transmission_spec_nights+trappist1_nights)*7
         print '\nTotal observing time for WP2 = %.1f hours (i.e. %.1f nights)'%(tot_time, tot_t
```

```
Measuring 1 temperature Earth-sized planet requires 41.5 hours (i.e. 5.9 nights).
Measuring 30 planets within the radius valley requires 276.4 hours (i.e. 39.5 nights).
Measuring 10 planets above the radius valley requires 49.9 hours (i.e. 7.1 nights).
Measuring 10 planets below the radius valley requires 128.2 hours (i.e. 18.3 nights).
Measuring 24 JWST follow-up planets requires 120.1 hours (i.e. 17.2 nights).
Total observing time for 75 TESS targets = 616.0 hours (i.e. 88.0 nights)

Transmission spectroscopy is allocated 35.0 hours (i.e. 5.0 nights)
Monitoring of the TRAPPIST-1 system is allocated 49.0 hours (i.e. 7.0 nights)

Total observing time for WP2 = 700.0 hours (i.e. 100.0 nights)
```

```
In [17]: fig = plt.figure(figsize=(9,6))
         ax1 = fig.add_subplot(111)
         g1 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med <= 1.5)
         ax1.hist(transmission_ppm[g1], bins=np.logspace(0,4.3,20), histtype='step', color='b',
         g3 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med <= 2) & (self.rps_med > 1.
         ax1.hist(transmission_ppm[g3], bins=np.logspace(0,4.3,20), histtype='step', color='k',
         g2 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med > 2)
         ax1.hist(transmission_ppm[g2], bins=np.logspace(0,4.3,20), histtype='step', color='r',
         ax1.hist(transmissionK2_ppm, bins=np.logspace(0,4.3,20), histtype='stepfilled', alpha=.
         ax1.set_xscale('log'), ax1.set_xlim((0,5e3)), ax1.legend(loc='upper left', fontsize=12)
         ax1.set_xlabel('Transmission signal (ppm)'), plt.ylabel('N_planets with a SPIRou mass f
         ax1.set_ylim((0,13))
```

Out[17]:

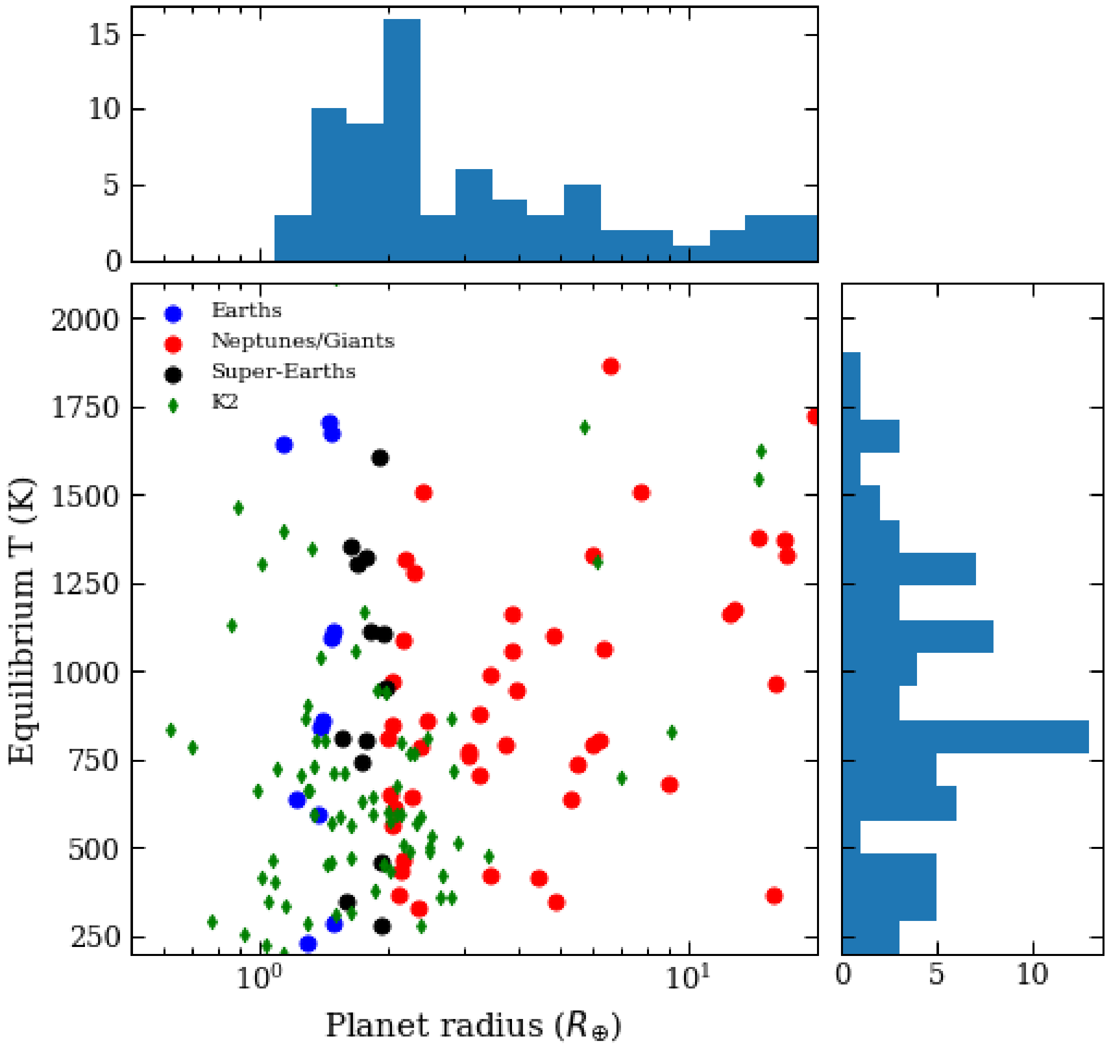
(0, 13)



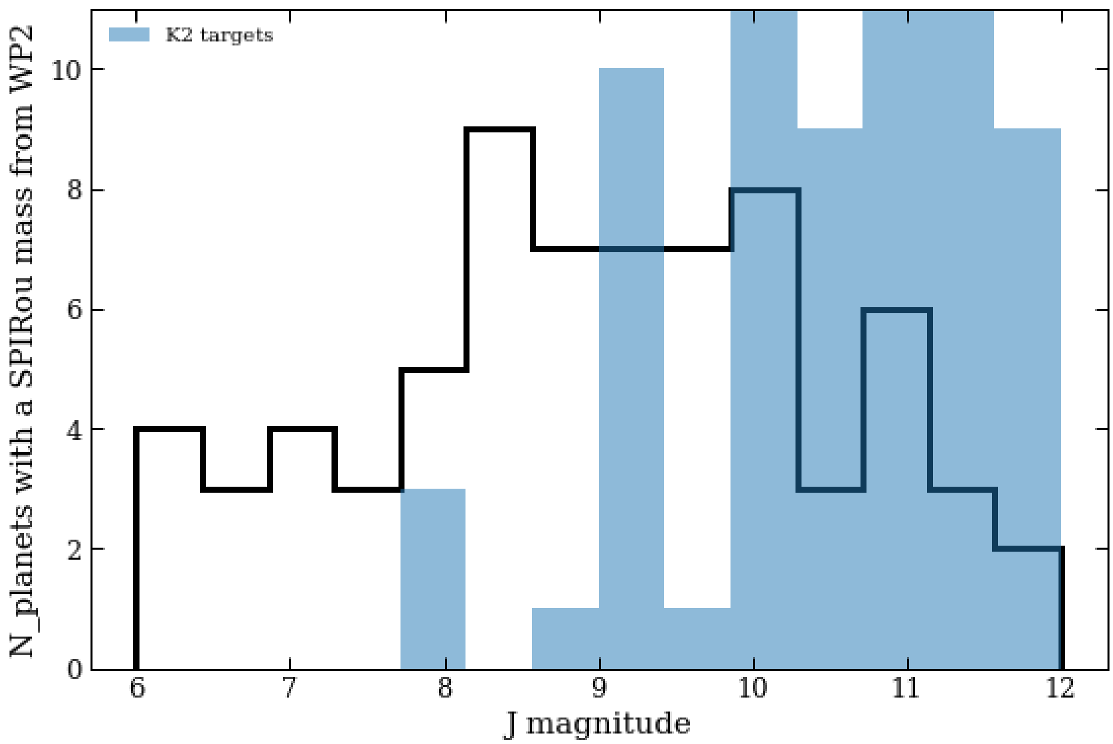

```
In [18]: fig = plt.figure(figsize=(8,8))
         gs = gridspec.GridSpec(7,7)
         ax1 = plt.subplot(gs[2:,:-2])
         ax2 = plt.subplot(gs[:2,:-2])
         ax3 = plt.subplot(gs[2:,-2:])
         ax1.scatter(self.rps_med[g1], Tps[g1], s=50, c='b', label='Earths'), ax1.set_xlim((.5,2
         ax1.scatter(self.rps_med[g2], Tps[g2], s=50, c='r', label='Neptunes/Giants'), ax1.set_x
         ax1.scatter(self.rps_med[g3], Tps[g3], s=50, c='k', label='Super-Earths')
         ax1.scatter(rpK2, TpK2, s=20, c='g', marker='d', label='K2')
         ax1.set_xlabel('Planet radius ($R_{\oplus}$)'), ax1.set_ylabel('Equilibrium T (K)')
         ax2.hist(self.rps_med[inds], bins=np.logspace(-.3,np.log10(20),20)), ax2.set_xscale('lo
         ax2.set_xticklabels(''), ax1.legend(loc='upper left')
         ax3.hist(Tps[inds], bins=np.linspace(2e2,2e3,20), orientation='horizontal')
         ax3.set_ylim((2e2,2.1e3)), ax3.set_yticklabels('')
```

Out[18]:

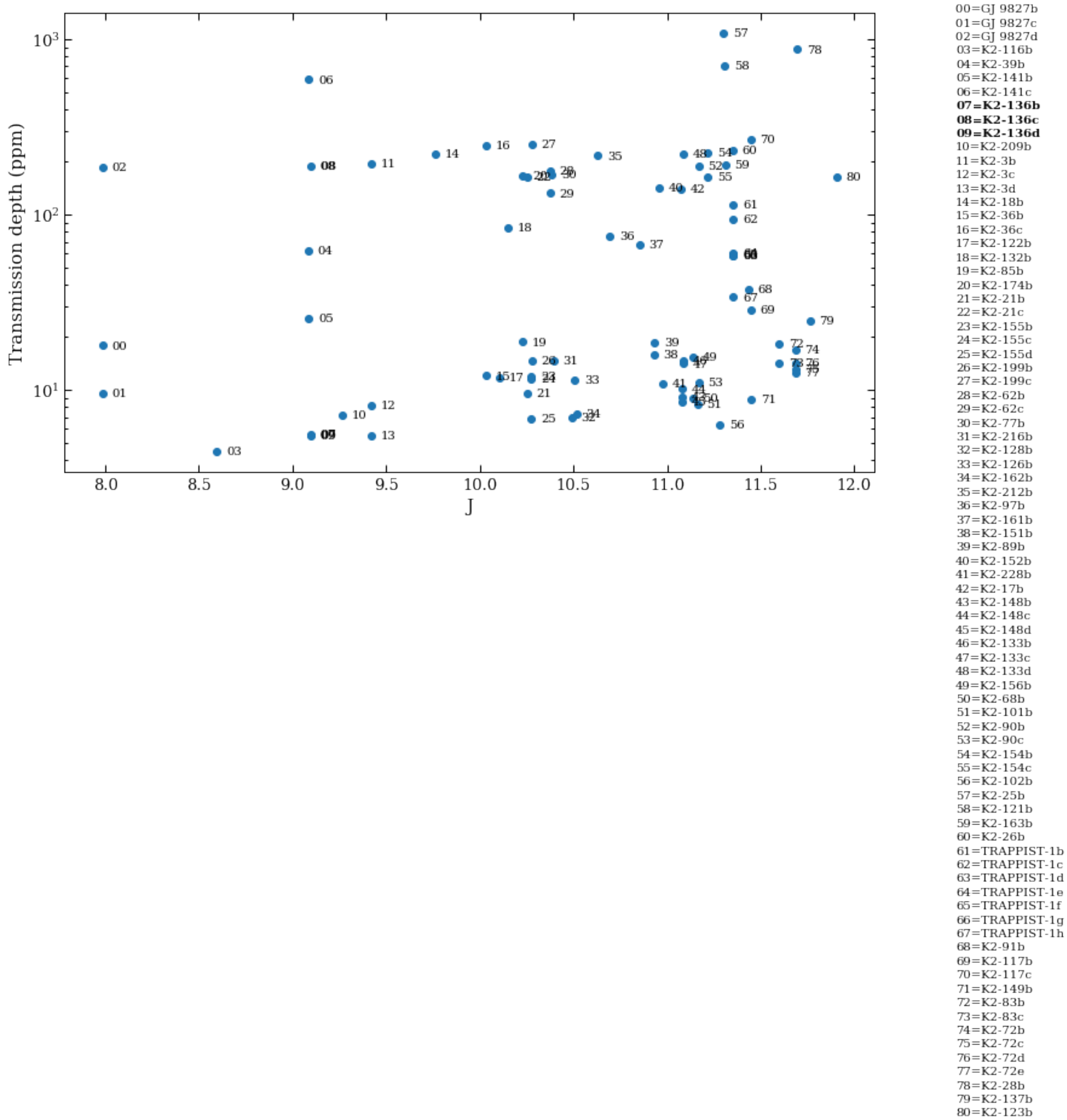
((200.0, 2100.0), [])




```
In [19]: fig = plt.figure(figsize=(9,6))
         ax1 = fig.add_subplot(111)
         ax1.hist(self.Jmags_med[inds], bins=np.linspace(6,12,15), histtype='step', color='k', l
         ax1.hist(JK2, bins=np.linspace(6,12,15), histtype='stepfilled', alpha=.5, label='K2 tar
         ax1.legend(loc='upper left')
         ax1.set_xlabel('J magnitude'), plt.ylabel('N_planets with a SPIRou mass from WP2'), ax1

Out[19]: (<matplotlib.text.Text at 0x1a16ceecd0>,
          <matplotlib.text.Text at 0x1a1639a090>,
          (0, 11))
```




```
In [20]: # K2 targets
         fig = plt.figure(figsize=(12,7))
         ax = fig.add_subplot(111)
         ax.scatter(JK2, transmissionK2_ppm), plt.yscale('log')
         ax.set_xlabel('J'), ax.set_ylabel('Transmission depth (ppm)')
         labels = np.arange(starnames.size)
         for i in range(starnames.size):
             weight = 'bold' if starname[i] == 'K2-136' else 'normal'
             ax.text(JK2[i]+.05, transmissionK2_ppm[i], '%.2d'%labels[i], verticalalignment='cen
             ax.text(1.1, 1-.03*i, '%.2d=%s'%(labels[i], starnames[i]), transform=ax.transAxes,
```


Transmission depth (ppm)
J
00=GJ 9827b
01=GJ 9827c
02=GJ 9827d
03=K2-116b
04=K2-39b
05=K2-141b
06=K2-141c
07=K2-136b
08=K2-136c
09=K2-136d
10=K2-209b
11=K2-3b
12=K2-3c
13=K2-3d
14=K2-18b
15=K2-36b
16=K2-36c
17=K2-122b
18=K2-132b
19=K2-85b
20=K2-174b
21=K2-21b
22=K2-21c
23=K2-155b
24=K2-155c
25=K2-155d
26=K2-199b
27=K2-199c
28=K2-62b
29=K2-62c
30=K2-77b
31=K2-216b
32=K2-128b
33=K2-126b
34=K2-162b
35=K2-212b
36=K2-97b
37=K2-161b
38=K2-151b
39=K2-89b
40=K2-152b
41=K2-228b
42=K2-17b
43=K2-148b
44=K2-148c
45=K2-148d
46=K2-133b
47=K2-133c
48=K2-133d
49=K2-156b
50=K2-68b
51=K2-101b
52=K2-90b
53=K2-90c
54=K2-154b
55=K2-154c
56=K2-102b
57=K2-25b
58=K2-121b
59=K2-163b
60=K2-26b
61=TRAPPIST-1b
62=TRAPPIST-1c
63=TRAPPIST-1d
64=TRAPPIST-1e
65=TRAPPIST-1f
66=TRAPPIST-1g
67=TRAPPIST-1h
68=K2-91b
69=K2-117b
70=K2-117c
71=K2-149b
72=K2-83b
73=K2-83c
74=K2-72b
75=K2-72c
76=K2-72d
77=K2-72e
78=K2-28b
79=K2-137b
80=K2-123b

```
In [ ]:
```