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

In [85]:
```
# write function that returns indices of planets of interest based on their
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 [86]:
```
# get K2 targets
fname = 'input_data/K2planets_Kdwarfs.csv'
starname, planetname = np.genfromtxt(fname,delimiter=',',skip_header=75,usec
starnames = np.array(['%s%s'%(starname[i], planetname[i]) for i in range(sta
PK2,aK2,rpK2,TeffK2,MsK2,RsK2,JK2 = np.genfromtxt(fname,delimiter=',',skip_h
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, T
```

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

In [87]:
```
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
```

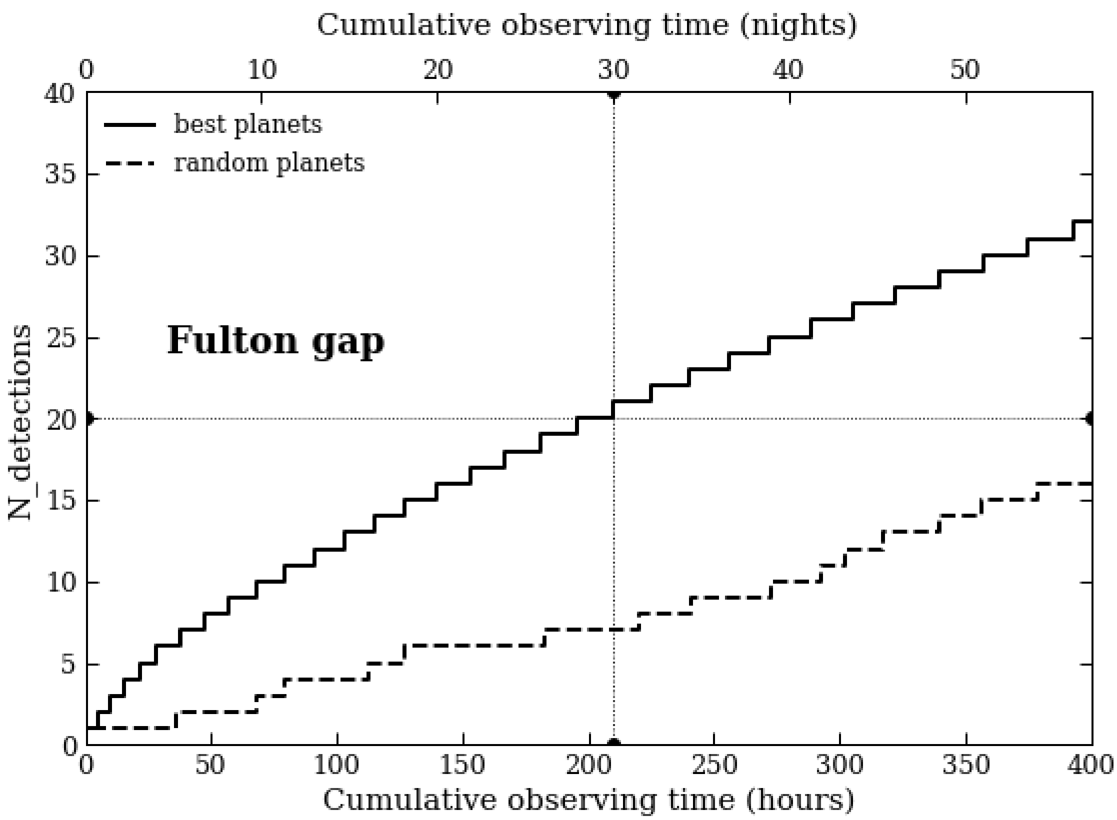
In [88]:
```
print 'Exposure time = %.2f minutes'%self.texps_med_N[ind]
print 'RV precision = %.2f m/s'%self.sigmaRV_phot_med_N[ind]
print 'Median effective RV rms = %.2f m/s'%self.sigmaRV_eff_med_N[ind]
print 'Median number of RV measurements required = %.1f'%self.NrvGPs_med_N[i
print 'Total observing time = %.1f hours (i.e. %.1f nights)'%(self.tobsGPs_m
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)
```

Measuring the 5σ mass of the 20 'best' planets around the Fulton gap

In [89]:
```
Nf1 = 20
scale = (.327/.189)**2  # 3 -> 5 sigma
g = (self.rps_med >= 1.5) & (self.rps_med <= 2) & (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', fo
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detec
ax1.text(.08, .6, 'Fulton gap', transform=ax1.transAxes, fontsize=18, weight
ax2 = ax1.twiny()
ax2.set_xlim((0,4e2/7)), ax2.set_xlabel('Cumulative observing time (nights)'
```

Out[89]:
```
((0, 57.142857142857146), <matplotlib.text.Text at 0x1a368a23d0>)
```

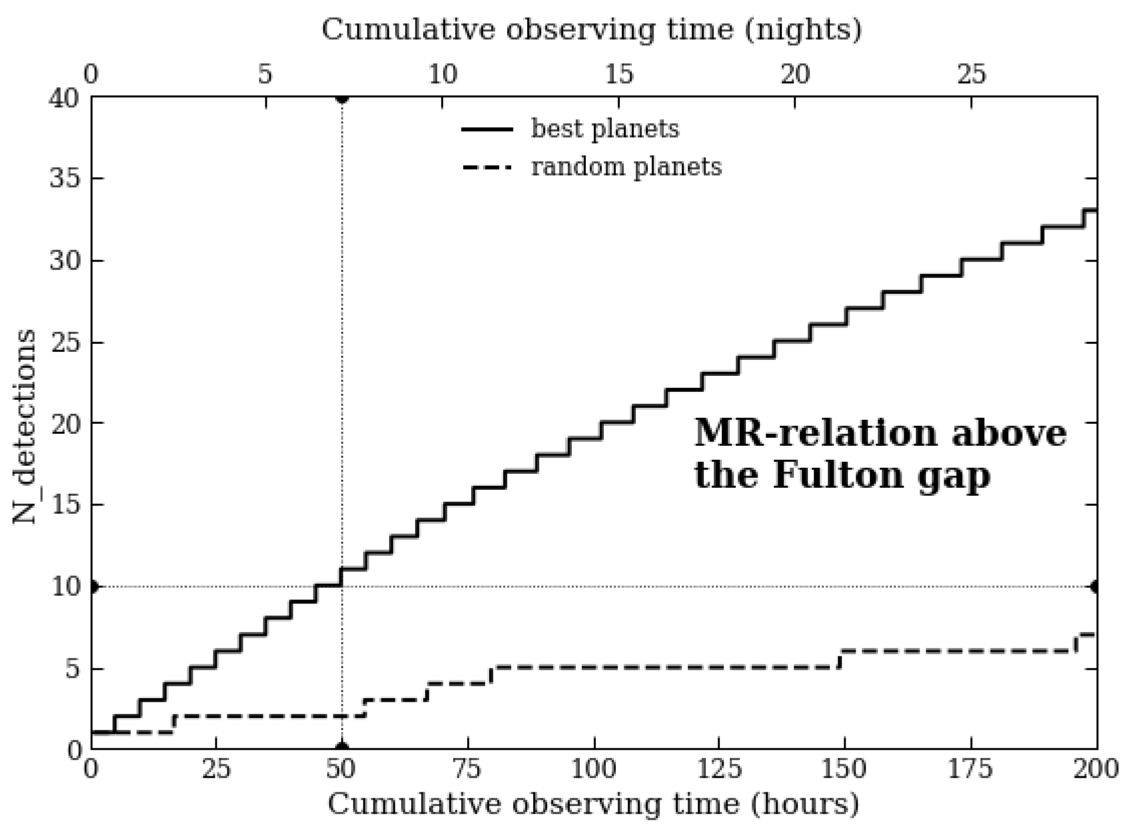


In [90]:
```
# save TOI indices
inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf1))
```

Extending the MR relation with 10 planets with $r_p \in [2, 4]R_\oplus$

In [91]:
```
Nf2 = 10
g = (self.rps_med > 2) & (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',
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detec
ax1.text(.6, .4, 'MR-relation above\nthe Fulton gap', transform=ax1.transAxe
ax2 = ax1.twiny()
ax2.set_xlim((0,2e2/7)), ax2.set_xlabel('Cumulative observing time (nights)'
```

Out[91]: ((0, 28.571428571428573), <matplotlib.text.Text at 0x1a208def10>)

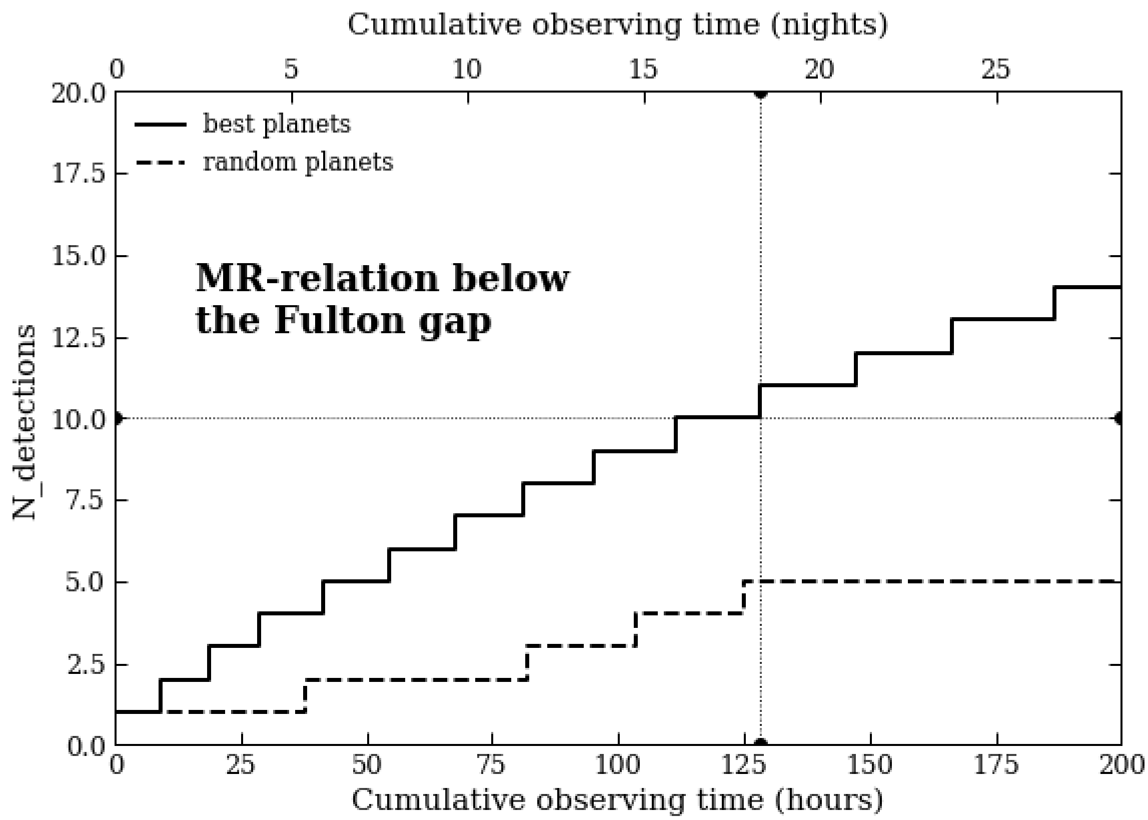


In [92]:
```
inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf2))
```

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

In [93]:
```
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', fo
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detec
ax1.text(.08, .63, 'MR-relation below\nthe Fulton gap', transform=ax1.transA
ax2 = ax1.twiny()
ax2.set_xlim((0,2e2/7)), ax2.set_xlabel('Cumulative observing time (nights)'
```

Out[93]: ((0, 28.571428571428573), <matplotlib.text.Text at 0x1a36b96710>)



In [94]:
```
inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf3))
```

DISREGARD** Measuring the 5σ masses TESS CVZ planets

In [95]:
```
# set number of WP2 nights available for RV follow-up
total_nights = 95
tobs_remaining = total_nights*7. - tobs_WP2.sum()
```

In [96]:

```
'''
g = (is_star_in_CVZ(self.ras_med, self.decs_med, 10).astype(bool)) & (self.c
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'
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(N
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detec
ax1.text(.08, .6, 'TESS CVZ', transform=ax1.transAxes, fontsize=18, weight='
ax2 = ax1.twiny()
ax2.set_xlim((0,np.ceil(tobs_remaining/1e2)*1e2/7)), ax2.set_xlabel('Cumulat
'''
```

Out[96]:

```
"\ng = (is_star_in_CVZ(self.ras_med, self.decs_med, 10).astype(bool)) &
(self.decs_med >= -20) & (np.in1d(np.arange(self.nstars), inds, invert=Tr
ue)) & (self.Jmags_med < 12)\ntobs = np.append(0, np.cumsum(np.sort(self.
tobsGPs_med_N[g]*scale)))\ntobs2 = np.append(0, np.cumsum(self.tobsGPs_me
d_N[g]*scale))\nNdet= np.arange(tobs.size)\nfig = plt.figure(figsize=(9,
6))\nax1 = fig.add_subplot(111)\nax1.plot(tobs, Ndet, 'k-', lw=2, drawsty
le='steps', label='best planets')\nax1.plot(tobs2, Ndet, 'k--', lw=2, dra
wstyle='steps', label='random planets')\nNjwst = int(Ndet[abs(tobs-tobs_r
emaining) == np.min(abs(tobs-tobs_remaining))])\ntobs_WP2 = np.append(tob
s_WP2, tobs_remaining)\nax1.axhline(Njwst, ls=':', lw=.8), ax1.axvline(to
bs_remaining, ls=':', lw=.9)\nax1.set_xlim((0,np.ceil(tobs_remaining/1e2)
*1e2)), ax1.set_ylim((0,np.ceil(Njwst/1e1)*1e1)), ax1.legend(loc='upper l
eft', fontsize=12)\nax1.set_xlabel('Cumulative observing time (hours)'),
ax1.set_ylabel('N_detections')\nax1.text(.08, .6, 'TESS CVZ', transform=a
x1.transAxes, fontsize=18, weight='semibold')\nax2 = ax1.twiny()\nax2.set
_xlim((0,np.ceil(tobs_remaining/1e2)*1e2/7)), ax2.set_xlabel('Cumulative
observing time (nights)', labelpad=12)\n"
```

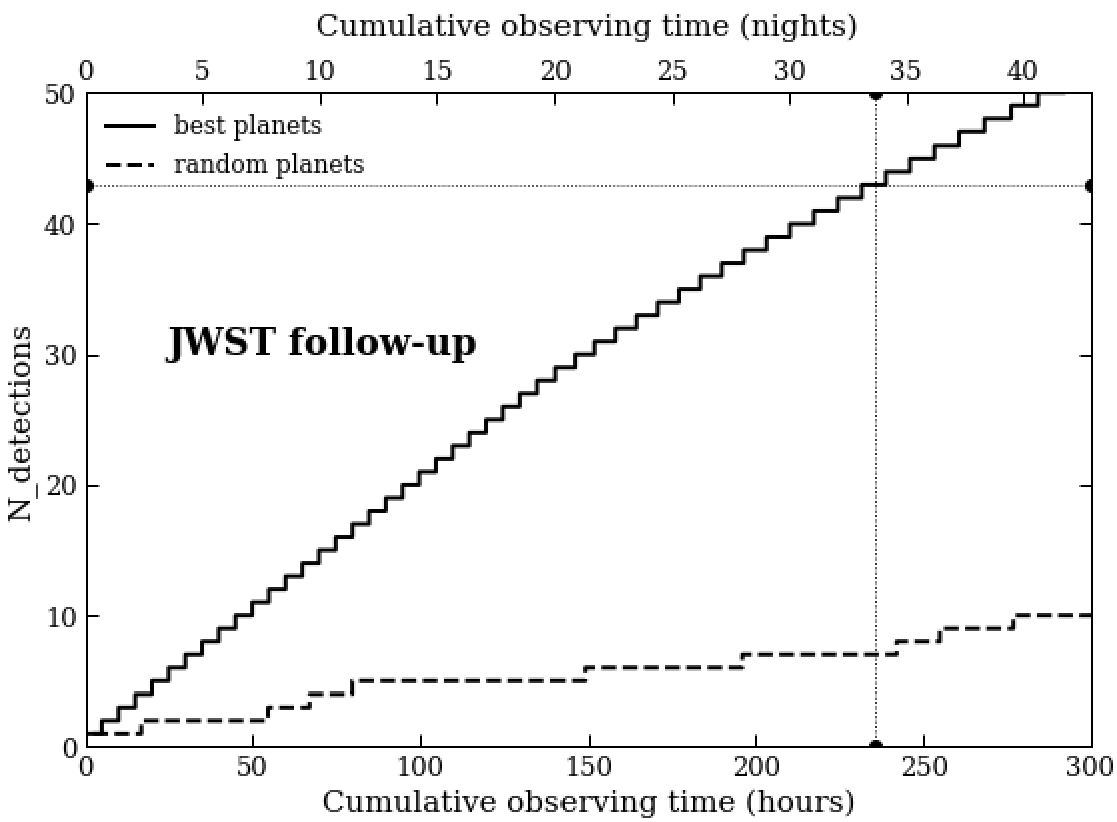
Measuring 5σ masses of TOIs amenable to transmission spectroscopy

In [97]:

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

In [98]:
```
gjwst = (np.in1d(np.arange(self.nstars), inds, invert=True)) & (self.Jmags_m
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(N
ax1.set_xlabel('Cumulative observing time (hours)'), ax1.set_ylabel('N_detec
ax1.text(.08, .6, 'JWST follow-up', transform=ax1.transAxes, fontsize=18, we
ax2 = ax1.twiny()
ax2.set_xlim((0,np.ceil(tobs_remaining/1e2)*1e2/7)), ax2.set_xlabel('Cumulat
```

Out[98]: ((0, 42.857142857142854), <matplotlib.text.Text at 0x1a1581f050>)



In [99]:
```
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 n
```

```
We detect 43 random potential JWST targets in 235.4 hours (i.e. 33.6 nigh
ts)
```

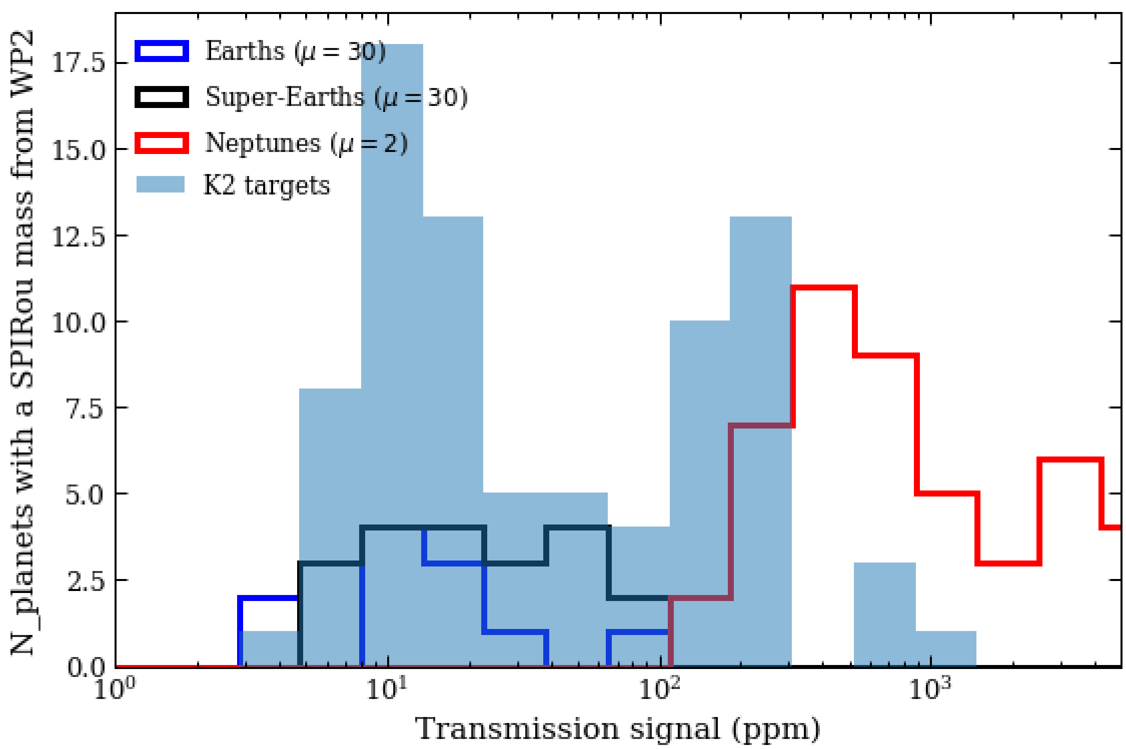
Summary of WP2 time allocations and planet populations

In [100]:
```
labels = ['1 temperature Earth-sized planet','%i planets at the Fulton gap'%
for i in range(tobs_WP2.size):
    print 'Measuring %s requires %.1f hours (i.e. %.1f nights).'%(labels[i],
print 'Total observing time for %i targets = %.1f hours (i.e. %.1f nights)'%
```

```
Measuring 1 temperature Earth-sized planet requires 41.5 hours (i.e. 5.9
nights).
Measuring 20 planets at the Fulton gap requires 210.0 hours (i.e. 30.0 ni
ghts).
Measuring 10 planets above the Fulton gap requires 49.9 hours (i.e. 7.1 n
ights).
Measuring 10 planets below the Fulton gap requires 128.2 hours (i.e. 18.3
nights).
Measuring 43 JWST follow-up planets requires 235.4 hours (i.e. 33.6 night
s).
Total observing time for 84 targets = 665.0 hours (i.e. 95.0 nights)
```

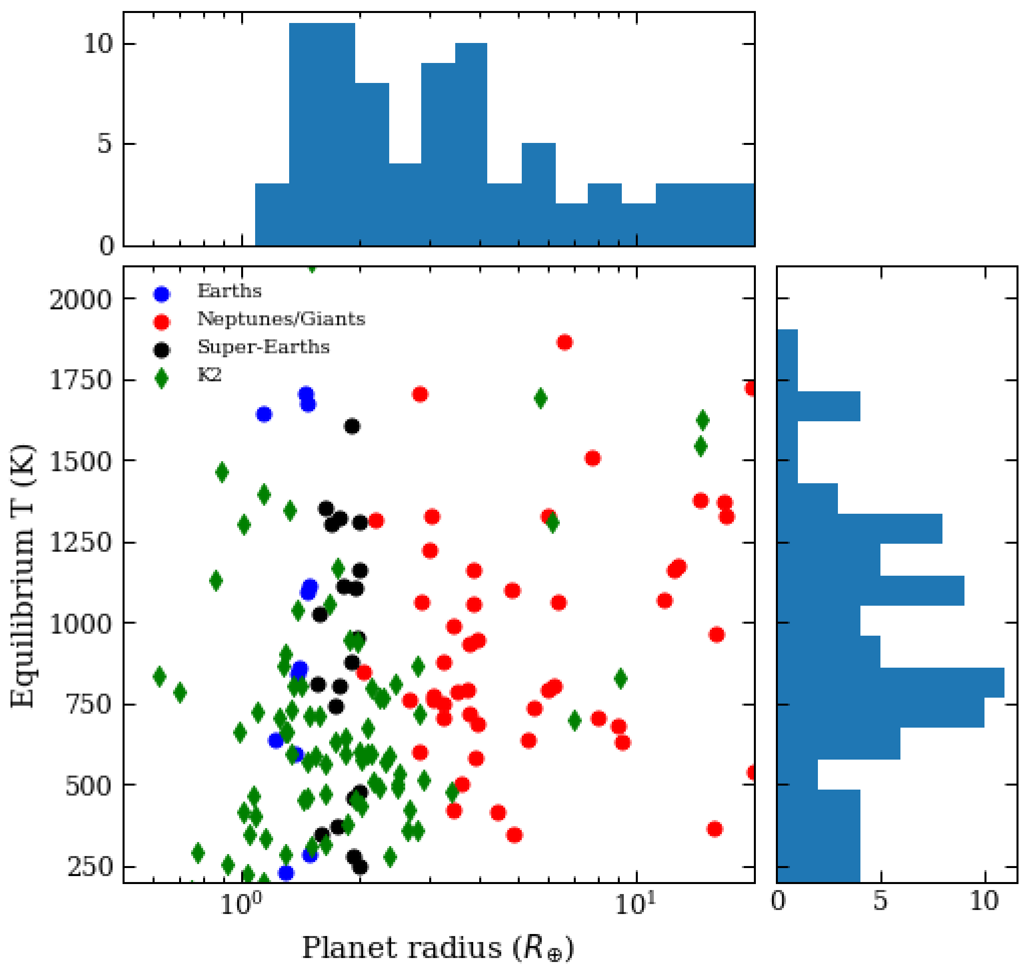
In [101]:
```
fig = plt.figure(figsize=(9,6))
ax1 = fig.add_subplot(111)
g1 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med <= 1.5)
ax1.hist(transmission_ppm[g1], bins=np.logspace(0,4.3,20), histtype='step',
g3 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med <= 2) & (self.r
ax1.hist(transmission_ppm[g3], bins=np.logspace(0,4.3,20), histtype='step',
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',
ax1.hist(transmissionK2_ppm, bins=np.logspace(0,4.3,20), histtype='stepfille
ax1.set_xscale('log'), ax1.set_xlim((0,5e3)), ax1.legend(loc='upper left', f
ax1.set_xlabel('Transmission signal (ppm)'), plt.ylabel('N_planets with a SP
```

Out[101]:
```
(<matplotlib.text.Text at 0x1a1656d3d0>,
 <matplotlib.text.Text at 0x1a1658cad0>)
```



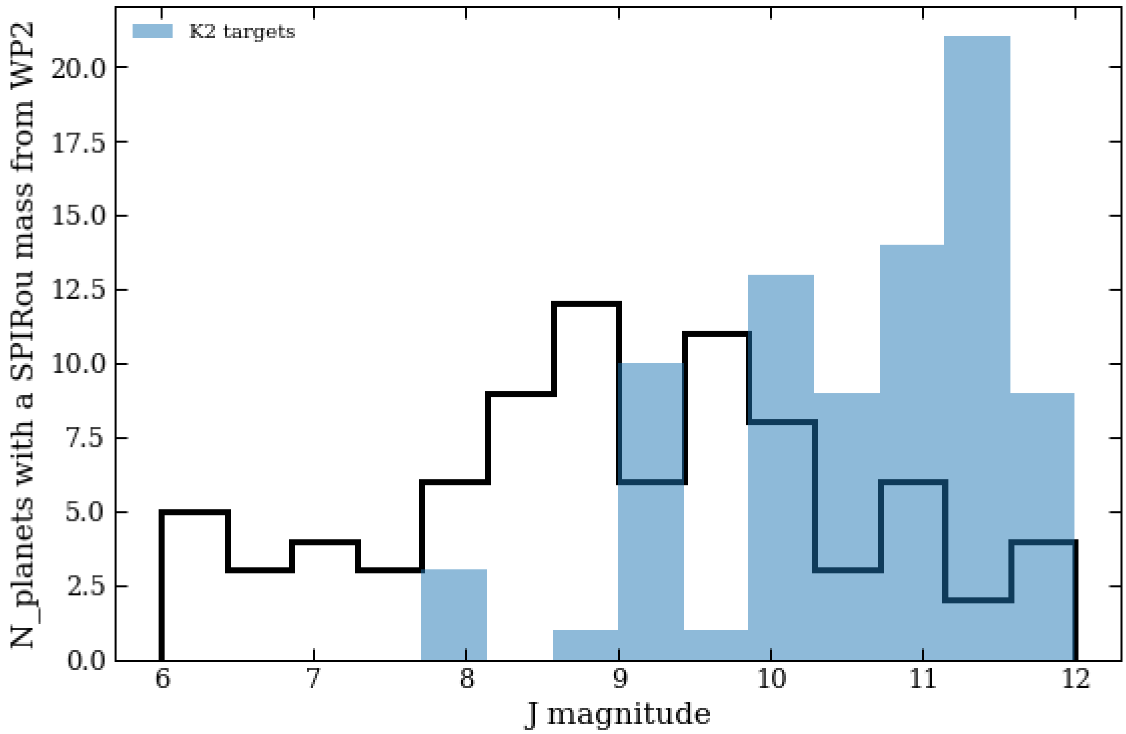
In [102]:
```
fig = plt.figure(figsize=(8,8))
gs = gridspec.GridSpec(7,7)
ax1 = plt.subplot(gs[2:,:-2])
ax2 = plt.subplot(gs[:2,:-2])
ax3 = plt.subplot(gs[2:,-2:])
ax1.scatter(self.rps_med[g1], Tps[g1], s=50, c='b', label='Earths'), ax1.set
ax1.scatter(self.rps_med[g2], Tps[g2], s=50, c='r', label='Neptunes/Giants')
ax1.scatter(self.rps_med[g3], Tps[g3], s=50, c='k', label='Super-Earths')
ax1.scatter(rpK2, TpK2, s=50, c='g', marker='d', label='K2')
ax1.set_xlabel('Planet radius ($R_{\oplus}$)'), ax1.set_ylabel('Equilibrium
ax2.hist(self.rps_med[inds], bins=np.logspace(-.3,np.log10(20),20)), ax2.set
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[102]: ((200.0, 2100.0), [])



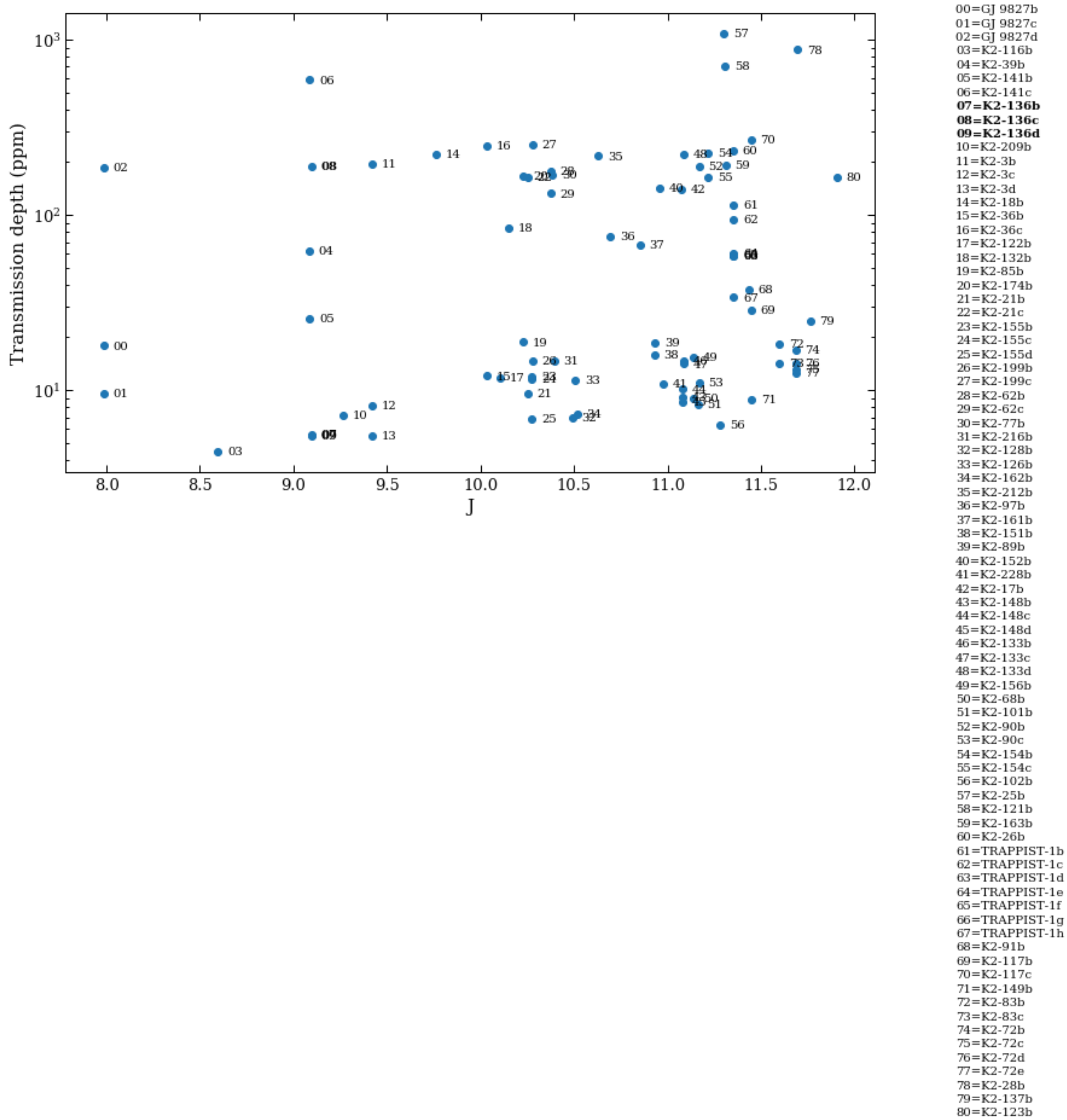
In [103]:
```
fig = plt.figure(figsize=(9,6))
ax1 = fig.add_subplot(111)
ax1.hist(self.Jmags_med[inds], bins=np.linspace(6,12,15), histtype='step', c
ax1.hist(JK2, bins=np.linspace(6,12,15), histtype='stepfilled', alpha=.5, la
ax1.legend(loc='upper left')
ax1.set_xlabel('J magnitude'), plt.ylabel('N_planets with a SPIRou mass fron
```

Out[103]:
```
(<matplotlib.text.Text at 0x1a36f143d0>,
 <matplotlib.text.Text at 0x1a1df7e490>)
```



In [104]:

```
# 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], verticalali
    ax.text(1.1, 1-.03*i, '%.2d=%s'%(labels[i], starnames[i]), transform=ax.
```



In [ ]: