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

In [412]:
```
# 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)]
```

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

In [413]:
```
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
```

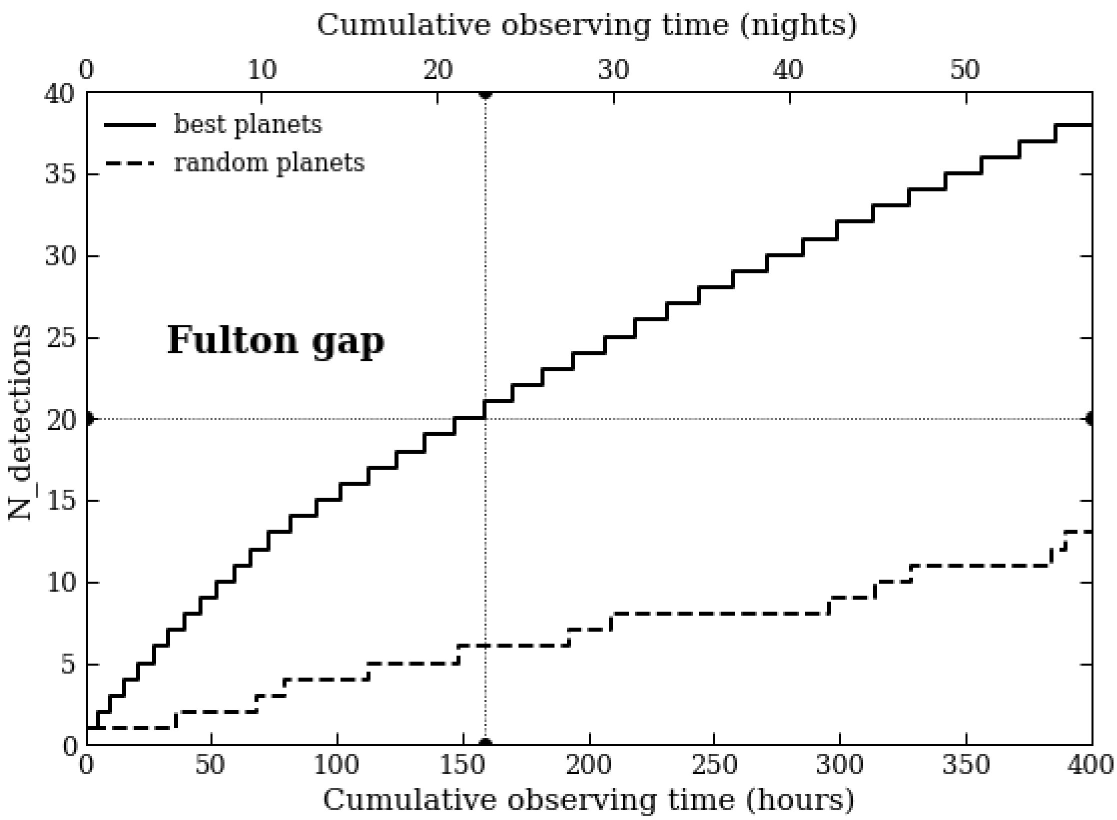
In [414]:
```
print 'Exposure time = %.2f minutes'%self.texps_med_N[ind]
print 'RV precision = %.2f m/s'%self.sigmaRV_phot_med_N[ind]
print 'Median effective RV rms = %.2f m/s'%self.sigmaRV_eff_med_N[ind]
print 'Median number of RV measurements required = %.1f'%self.NrvGPs_med_N[i
print 'Total observing 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 30 'best' planets around the Fulton gap

In [415]:
```
Nf1 = 20
scale = (.327/.189)**2  # 3 -> 5 sigma
g = (self.rps_med >= 1.5) & (self.rps_med <= 2)
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[415]: ((0, 57.142857142857146), <matplotlib.text.Text at 0x1a279e7390>)

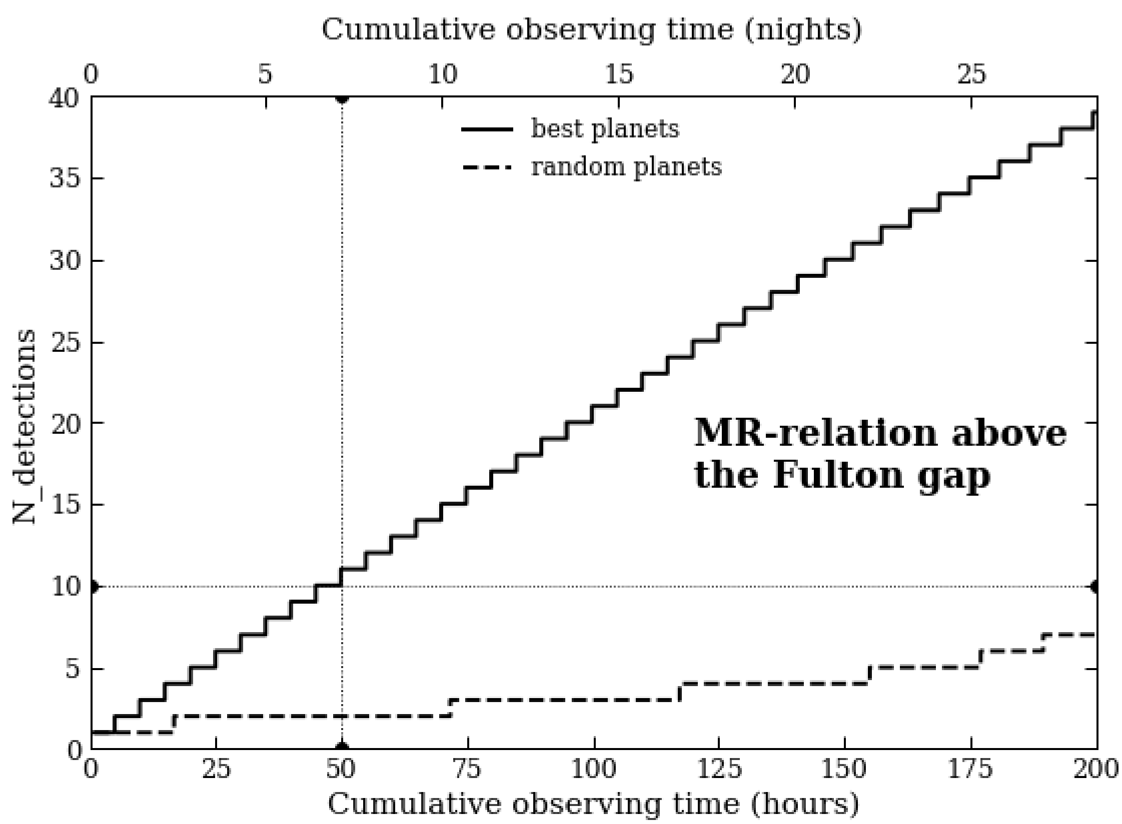


In [416]:
```
# save TOI indices
inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf1))
```

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

In [417]:
```
Nf2 = 10
g = (self.rps_med > 2) & (self.rps_med <= 4)
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[417]:
```
((0, 28.571428571428573), <matplotlib.text.Text at 0x1a28d1b950>)
```

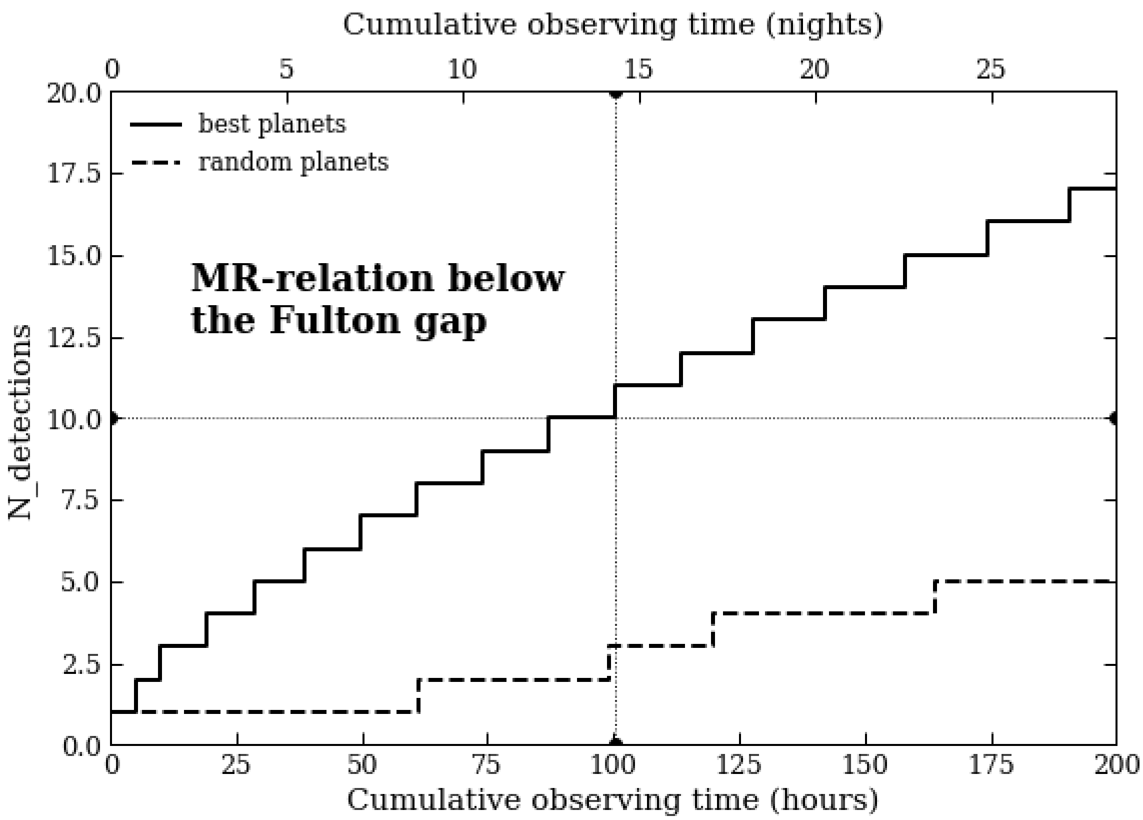


In [418]:
```
inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf2))
```

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

In [419]:
```
Nf3=10
g = (self.rps_med <= 1.5)
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[419]: ((0, 28.571428571428573), <matplotlib.text.Text at 0x1a28b43e90>)



In [420]:
```
inds = np.append(inds, get_planets(self.tobsGPs_med_N, g, Nf3))
```

DISREGARD** Measuring the 5σ masses TESS CVZ planets

In [421]:
```
# set number of WP2 nights available for RV follow-up
total_nights = 95
tobs_remaining = total_nights*7. - tobs_WP2.sum()
```

In [422]:
```
'''
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
'''
```

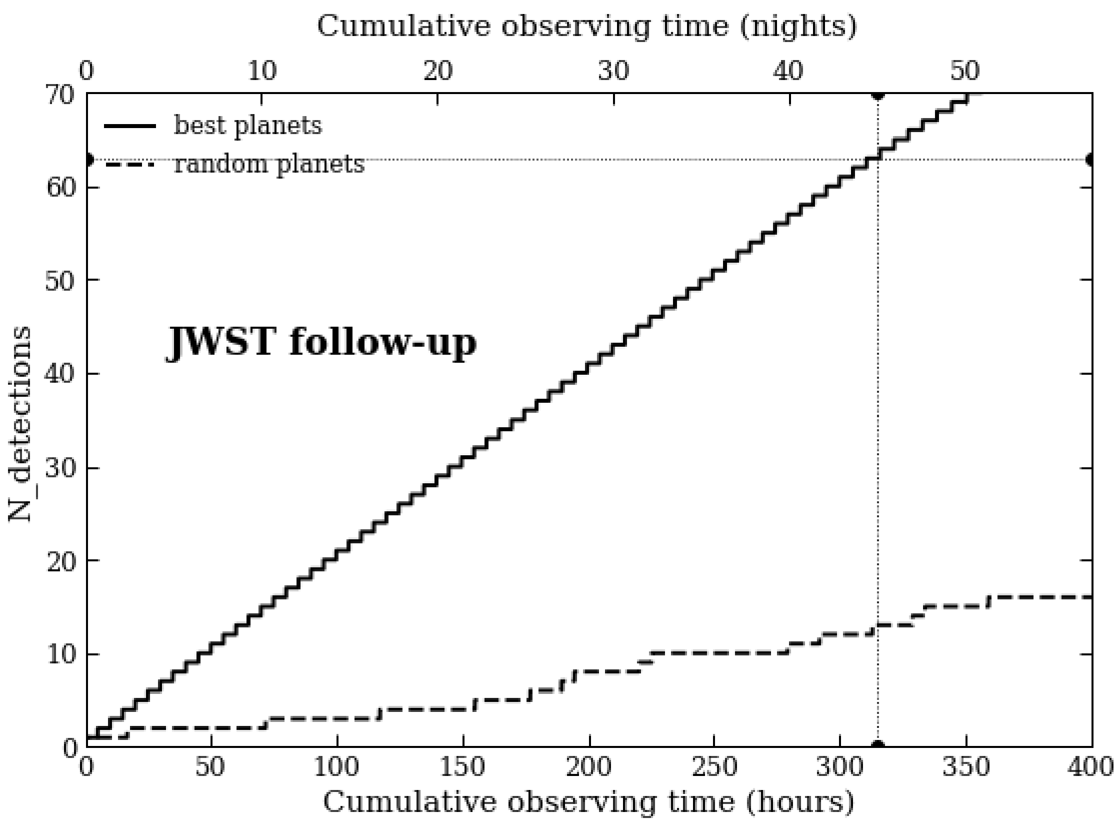
Out[422]:
```
"\ng = (is_star_in_CVZ(self.ras_med, self.decs_med, 10).astype(bool)) &
(self.decs_med >= -20) & (np.in1d(np.arange(self.nstars), inds, invert=Tr
ue)) & (self.Jmags_med < 12)\ntobs = np.append(0, np.cumsum(np.sort(self.
tobsGPs_med_N[g]*scale)))\ntobs2 = np.append(0, np.cumsum(self.tobsGPs_me
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 [423]:
```
# 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 [424]:
```
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[424]: ((0, 57.142857142857146), <matplotlib.text.Text at 0x1a41fcfb10>)



In [425]:
```
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 r
```

```
We detect 63 random potential JWST targets in 314.9 hours (i.e. 45.0 nigh
ts)
```

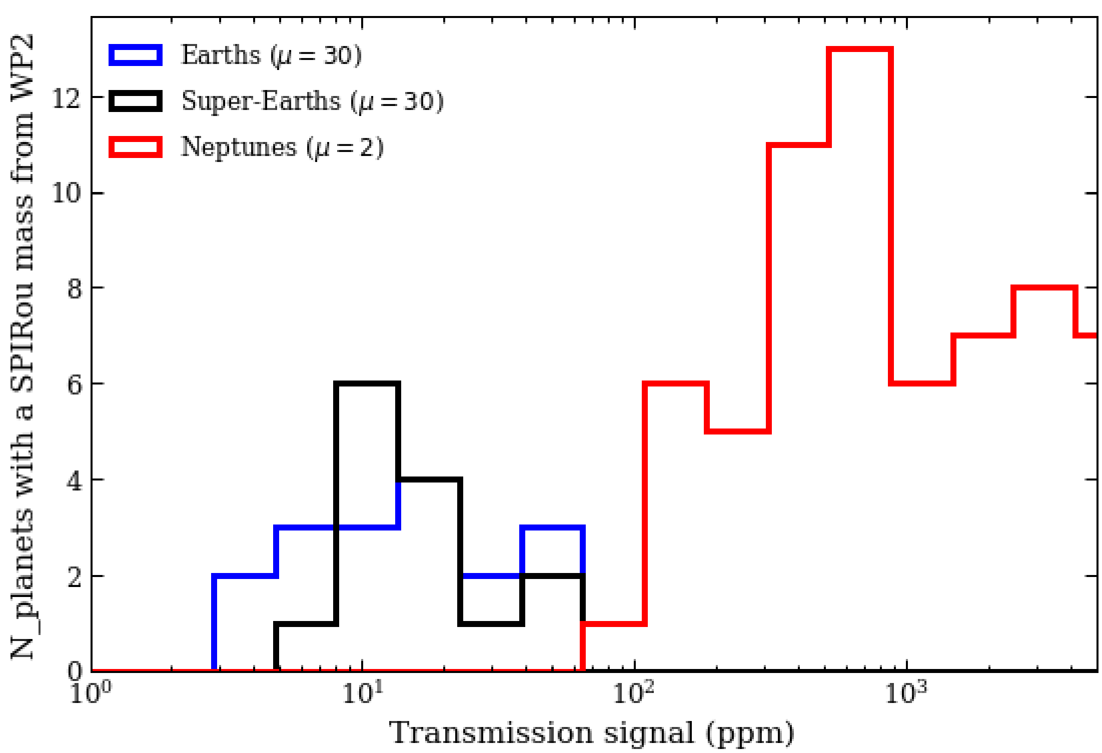
Summary of WP2 time allocations and planet populations

In [426]:
```
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 158.6 hours (i.e. 22.7 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 100.2 hours (i.e. 14.3
nights).
Measuring 63 JWST follow-up planets requires 314.9 hours (i.e. 45.0 night
s).
Total observing time for 104 targets = 665.0 hours (i.e. 95.0 nights)
```

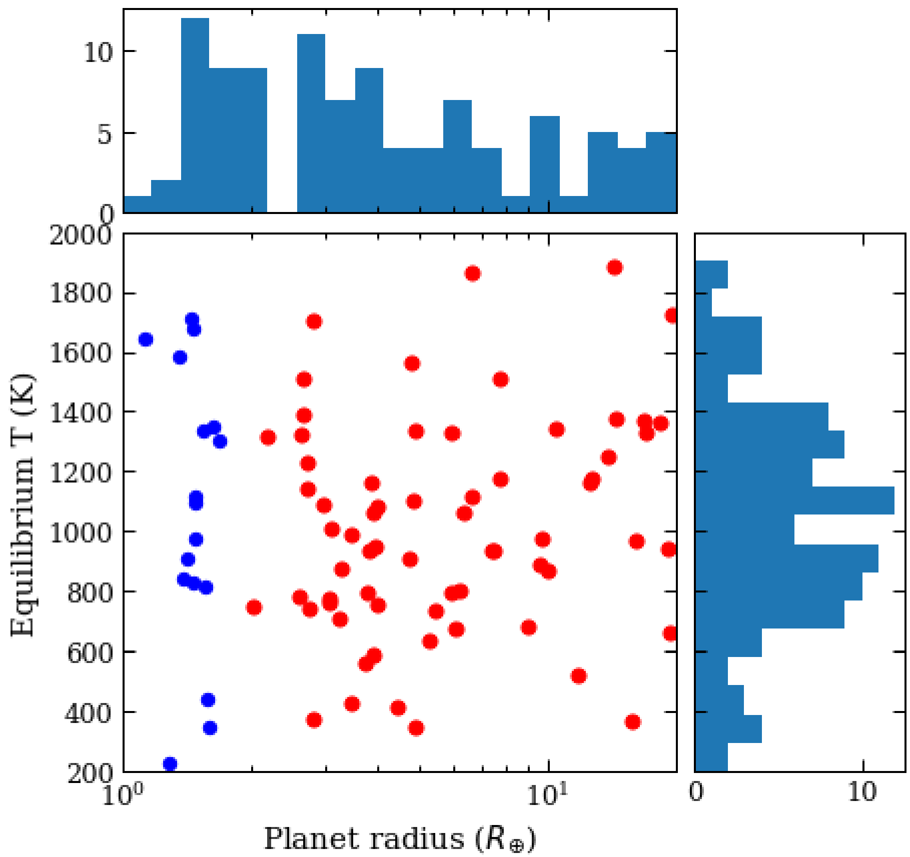
In [427]:
```
fig = plt.figure(figsize=(9,6))
ax1 = fig.add_subplot(111)
g1 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med <= 1.7)
ax1.hist(transmission_ppm[g1], bins=np.logspace(0,4.3,20), histtype='step',
g3 = (np.in1d(np.arange(self.nstars), inds)) & (self.rps_med <= 2) & (self.r
ax1.hist(transmission_ppm[g3], 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.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[427]:
```
(<matplotlib.text.Text at 0x1a41f81510>,
 <matplotlib.text.Text at 0x1a3fc63810>)
```



In [428]:
```
fig = plt.figure(figsize=(7,7))
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=40, c='b'), ax1.set_xlim((1,20)), a
ax1.scatter(self.rps_med[g2], Tps[g2], s=50, c='r'), ax1.set_xscale('log')
ax1.set_xlabel('Planet radius ($R_{\oplus}$)'), ax1.set_ylabel('Equilibrium
ax2.hist(self.rps_med[inds], bins=np.logspace(0,np.log10(20),20)), ax2.set_x
ax2.set_xticklabels('')
ax3.hist(Tps[inds], bins=np.linspace(2e2,2e3,20), orientation='horizontal')
ax3.set_ylim((2e2,2e3)), ax3.set_yticklabels('')
```

Out[428]: ((200.0, 2000.0), [])



In [429]:
```
fig = plt.figure(figsize=(9,6))
ax1 = fig.add_subplot(111)
ax1.hist(self.Jmags_med[inds], bins=15, histtype='step', color='k', lw=3)
ax1.set_xlabel('J magnitude'), plt.ylabel('N_planets with a SPIRou mass fron
```

Out[429]:
```
(<matplotlib.text.Text at 0x1a3ea65890>,
 <matplotlib.text.Text at 0x1a3f6d3e50>)
```