In [66]:
```
train.head()
```

Out[66]:

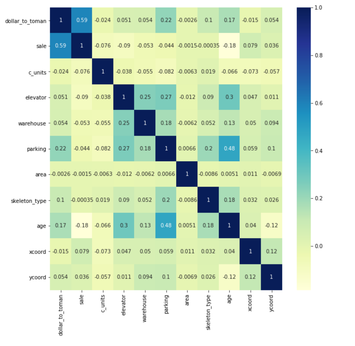
	dollar_to_toman	sale	c_units	elevator	warehouse	parking	area	skeleton_type	age	xcoord	ycoord
5433	1660000.0	11840000	19.0	0	0	0	58.58	0	1	5708828.884	4265513.78
6488	2900000.0	18000000	2.0	1	1	1	76.00	0	24	5710637.904	4268569.86
2840	145500.0	9780000	4.0	0	0	0	62.41	0	13	5712940.368	4262458.18
4557	138490.0	18350000	7.0	0	0	0	84.35	0	8	5710763.546	4264962.93
2389	138460.0	21500000	2.0	0	0	0	130.35	1	8	5713750.557	4264698.95

In [65]:
```
# import modules
import matplotlib.pyplot as mp
import seaborn as sb

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

# plotting correlation heatmap
dataplot = sb.heatmap(train.corr(), cmap="YlGnBu", annot=True)

# displaying heatmap
mp.show()
```



In [61]:
```
# VIF dataframe
vif_data = pd.DataFrame()
train_copy=train.copy()
train_copy=train_copy.drop(columns=['sale','xcoord','ycoord'])
vif_data["feature"] = train_copy.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(train_copy.values, i)
                   for i in range(len(train_copy.columns))]

print(vif_data)
```

```
          feature       VIF
0  dollar_to_toman  2.696573
1          c_units  1.239813
2         elevator  1.272388
3       warehouse  1.279663
4          parking  1.601623
5             area  1.006752
6    skeleton_type  1.219485
7              age  2.998644
```

In [39]:
```
tr_y = train['sale'].values

tr_X = train[['dollar_to_toman', 'c_units', 'elevator', 'warehouse', 'parking', 'area', 'skeleton_type', 'age']].values

utr = train['xcoord']
vtr = train['ycoord']
tr_coords = list(zip(utr,vtr))

tr_X = (tr_X - tr_X.mean(axis=0)) / tr_X.std(axis=0)

tr_y = tr_y.reshape((-1,1))

tr_y = (tr_y - tr_y.mean(axis=0)) / tr_y.std(axis=0)
```

In [53]:
```
#Calibrate GWR model

mgwr_selector = Sel_BW(coords=tr_coords, y=tr_y, X_loc=tr_X, multi=True, kernel='gaussian', fixed=True)
mgwr_bw = mgwr_selector.search(verbose=True,criterion='AICc')
print(gwr_bw)
```

```
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=8.73625e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=7.73224e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=1.50723e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=1.36652e-17): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=2.04209e-17): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=5.79554e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=1.0794e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=1.82269e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=5.57826e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=6.44526e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=2.43056e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=5.96317e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=4.97844e-19): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=4.18497e-29): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=5.84169e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=2.60174e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=2.1558e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
/usr/local/lib/python3.7/dist-packages/spglm/iwls.py:37: LinAlgWarning: Ill-conditioned matrix (rcond=4.29049e-18): result may not be accurate.
  xtx_inv_xt = linalg.solve(xtx, xT)
```

```
---------------------------------------------------------------------------
LinAlgError                               Traceback (most recent call last)
<ipython-input-53-0ef7a5164beb> in <module>()
      2
      3 mgwr_selector = Sel_BW(coords=tr_coords, y=tr_y, X_loc=tr_X, multi=True, kernel='gaussian', fixed=True)
----> 4 mgwr_bw = mgwr_selector.search(verbose=True,criterion='AICc')
      5 print(gwr_bw)

/usr/local/lib/python3.7/dist-packages/mgwr/sel_bw.py in search(self, search_method, criterion, bw_min, bw_max, interval, tol, max_iter, init_multi, tol_multi, rss_score, max_iter_multi, multi_bw_min, multi_bw_max, bws_same_times, pool, verbose)
    311
    312             if self.multi:
--> 313                 self._mbw()
    314                 self.params = self.bw[3]  #params n by k
    315                 self.sel_hist = self.bw[-2] #bw searching history

/usr/local/lib/python3.7/dist-packages/mgwr/sel_bw.py in _mbw(self)
    400                                 self.max_iter_multi, self.rss_score, gwr_func,
    401                                 bw_func, sel_func, multi_bw_min, multi_bw_max,
--> 402                                 bws_same_times, verbose=self.verbose)
    403
    404     def _init_section(self, X_glob, X_loc, coords, constant):

/usr/local/lib/python3.7/dist-packages/mgwr/search.py in multi_bw(init, y, X, n, k, family, tol, max_iter, rss_score, gwr_func, bw_func, sel_func, multi_bw_min, multi_bw_max, bws_same_times, verbose)
    180     """
    181     if init is None:
--> 182         bw = sel_func(bw_func(y, X))
    183         optim_model = gwr_func(y, X, bw)
    184     else:

/usr/local/lib/python3.7/dist-packages/mgwr/sel_bw.py in sel_func(bw_func, bw_min, bw_max)
    395                 search_method=search_method, criterion=criterion,
    396                 bw_min=bw_min, bw_max=bw_max, interval=interval, tol=tol,
--> 397                 max_iter=max_iter, pool=self.pool, verbose=False)
    398
    399         self.bw = multi_bw(self.init_multi, y, X, n, k, family, self.tol_multi,

/usr/local/lib/python3.7/dist-packages/mgwr/sel_bw.py in search(self, search_method, criterion, bw_min, bw_max, interval, tol, max_iter, init_multi, tol_multi, rss_score, max_iter_multi, multi_bw_min, multi_bw_max, bws_same_times, pool, verbose)
    317                 -1]  #scalar, optimal bw from initial gwr model
    318         else:
--> 319             self._bw()
    320             self.sel_hist = self.bw[-1]
    321

/usr/local/lib/python3.7/dist-packages/mgwr/sel_bw.py in _bw(self)
    337             self.bw = golden_section(a, c, delta, gwr_func, self.tol,
    338                                      self.max_iter, self.int_score,
--> 339                                      self.verbose)
    340         elif self.search_method == 'interval':
    341             self.bw = equal_interval(self.bw_min, self.bw_max, self.interval,

/usr/local/lib/python3.7/dist-packages/mgwr/search.py in golden_section(a, c, delta, functio
n, tol, max_iter, int_score, verbose)
     60             score_b = dict[b]
     61         else:
---> 62             score_b = function(b)
     63             dict[b] = score_b
     64             if verbose:

/usr/local/lib/python3.7/dist-packages/mgwr/sel_bw.py in <lambda>(bw)
    327             self.coords, self.y, self.X_loc, bw, family=self.family, kernel=
    328             self.kernel, fixed=self.fixed, constant=self.constant, offset=self.
--> 329             offset, spherical=self.spherical).fit(lite=True, pool=self.pool)}
    330
    331         self._optimized_function = gwr_func

/usr/local/lib/python3.7/dist-packages/mgwr/gwr.py in fit(self, ini_params, tol, max_iter, solve, lite, pool)
    333                     rslt = map(self._local_fit, range(m))  #sequential
    334
--> 335             rslt_list = list(zip(*rslt))
    336             influ = np.array(rslt_list[0]).reshape(-1, 1)
    337             resid = np.array(rslt_list[1]).reshape(-1, 1)

/usr/local/lib/python3.7/dist-packages/mgwr/gwr.py in _local_fit(self, i)
    249
    250         if isinstance(self.family, Gaussian):
--> 251             betas, inv_xtx_xt = _compute_betas_gwr(self.y, self.X, wi)
    252             predy = np.dot(self.X[i], betas)[0]
    253             resid = self.y[i] - predy

/usr/local/lib/python3.7/dist-packages/spglm/iwls.py in _compute_betas_gwr(y, x, wi)
     35     xT = (x * wi).T
     36     xtx = np.dot(xT, x)
---> 37     xtx_inv_xt = linalg.solve(xtx, xT)
     38     betas = np.dot(xtx_inv_xt, y)
     39     return betas, xtx_inv_xt

/usr/local/lib/python3.7/dist-packages/scipy/linalg/basic.py in solve(a, b, sym_pos, lower, overwrite_a, overwrite_b, debug, check_finite, assume_a, transposed)
    214                                  (a1, b1))
    215         lu, ipvt, info = getrf(a1, overwrite_a=overwrite_a)
--> 216         _solve_check(n, info)
    217         x, info = getrs(lu, ipvt, b1,
    218                         trans=trans, overwrite_b=overwrite_b)

/usr/local/lib/python3.7/dist-packages/scipy/linalg/basic.py in _solve_check(n, info, lamch, rcond)
     29                                  '.'.format(-info))
     30     elif 0 < info:
---> 31         raise LinAlgError('Matrix is singular.')
     32
     33     if lamch is None:

LinAlgError: Matrix is singular.
```