Курсовая работа по теории вероятностей и математической статистике

- Студент: Перевозчиков Георгий
- Группа: М8О-305Б-17
- Руководитель: Семенихин К. В

Задача 1

1. Смоделировать движение самолета в плоской декартовой системе координат Oxy и процесс его наблюдения наземным измерительным средством (НИС). Самолет движется прямолинейно и равномерно со скоростью $100 \div 300$ м/с на расстоянии $10 \div 200$ км от НИС, расположенного в начале координат Oxy. Проводятся $n = 20 \div 40$ измерений $\{Y_k\}$ дальности $r = \sqrt{x^2 + y^2}$ с постоянным временным шагом $h = 2 \div 5$ с. Ошибки наблюдения не содержат систематической погрешности и образуют набор независимых случайных величин $\{W_k\}$, распределенных по нормальному закону с одинаковой дисперсией σ^2, где $\sigma = 100 \div 1000$ м.

Задать уровень значимости $\alpha = 0{,}01 \div 0{,}005$.

Решение

1. Определим вариативные параметры и импортируем необходимые библиотеки:

- `n` - колличкство измерений
- `h` - временной шаг
- `v` - линейная скорость
- `sigma` - среднеквадратическое отклонение
- `disp` - дисперсия
- `y0` - начальное расстояние от наблюдателя (самолет над наблюдателем)
- `t` - вектор времени: $t_i = h * i$
- `x` - вектор координат самолета в каждый момент времени t: $x_i = v * t_t$
- `r` - вектор дальностьи а каждый момент времени
- `norm_errors` - ошибки наблюдения: их можно сгенерировать с помошью `np.random.normal(0, sigma)`
- `alpha` - уровень значимости

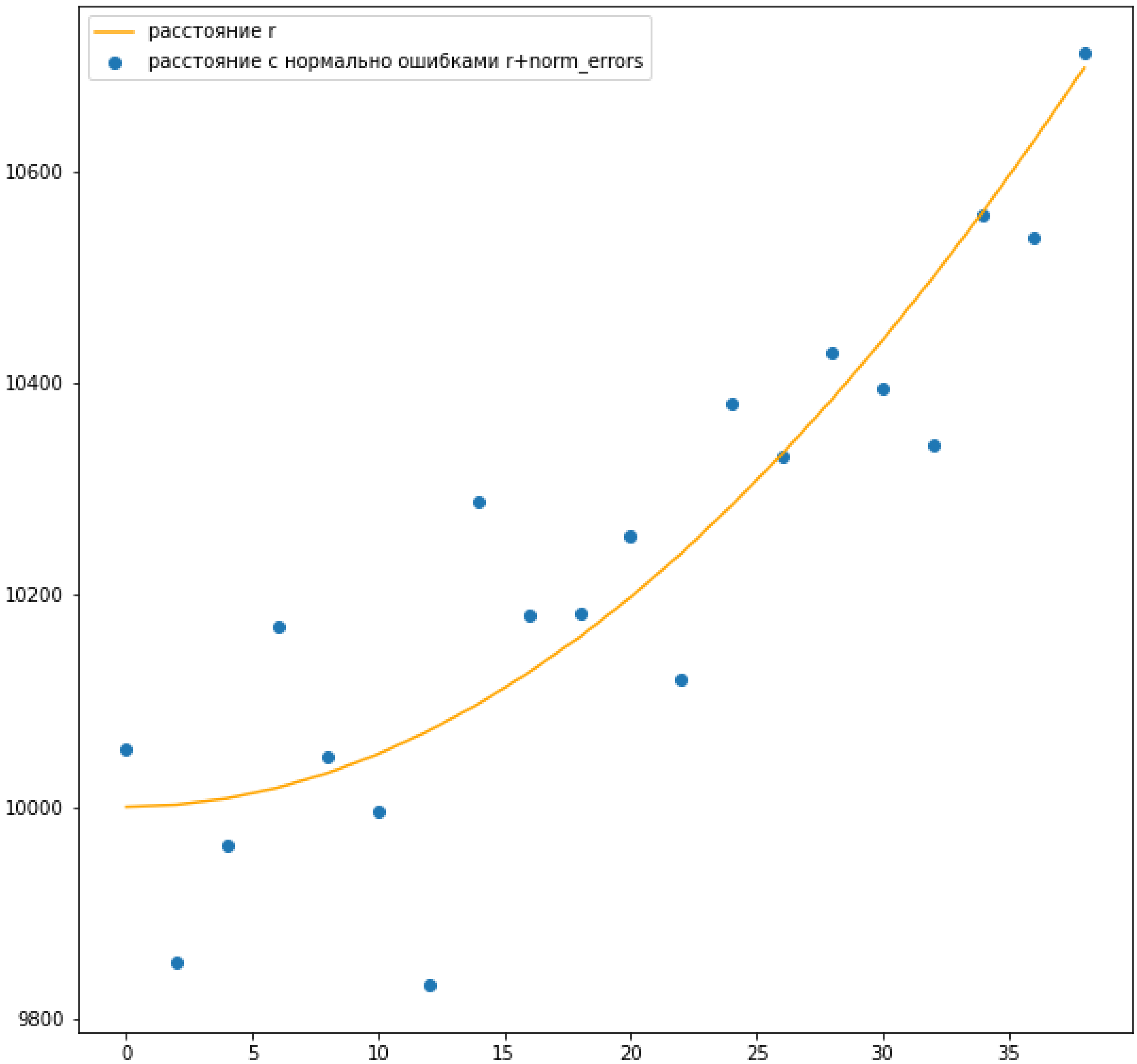
In [1]:
```
# импорт библиотек
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
# вариативные параметры
n = 20
h = 2
v = 100
sigma = 100
```

```
disp = sigma**2
y0 = 10000
t = np.arange(0, h*n, h)
x = v*t
r = np.sqrt(x**2 + y0**2)
norm_errors = np.random.normal(0, sigma, t.shape[0])
alpha = 0.01
```

1. Построим график

In [3]:
```
plt.figure(figsize=(10,10))
plt.plot(t, r, label = "расстояние r", color= "orange")
plt.scatter(t, r+norm_errors, label = "расстояние с нормально ошибками r+norm_er
plt.legend()
plt.show()
```



1. постотрим на вектор расстояния и расстояния с ошибками

In [4]:
```
print('расстояние:\n', r)
print('расстояние с ошибками:\n', r+norm_errors)
```

```
расстояние:
 [10000.         10001.99980004 10007.99680256 10017.98382909
```

```
 10031.94896319 10049.87562112 10071.74264961 10097.52444909
 10127.19112094 10160.7086367  10198.03902719 10239.14058894
 10283.96810575 10332.47308247 10384.6039886  10440.30650891
 10499.52379873 10562.19674121 10628.26420447 10697.66329625]
расстояние с ошибками:
 [10054.4185668   9853.76284322  9964.49142514 10171.18985532
 10047.98334741  9996.39222432  9831.71965145 10287.11503691
 10181.41364269 10183.26181203 10255.0070167  10120.45837768
 10379.65089657 10331.23327909 10427.88464285 10394.63628785
 10341.64344827 10558.75466767 10537.00432824 10711.98920701]
```

Задача 2

2. Используя метод наименьших квадратов (МНК), оценить параметры модели, считая, что наблюдения описываются моделью простой линейной регрессии

$$Y_k = \theta_1 + \theta_2 t_k + W_k, \quad k = 1, \ldots, n,$$

где $\{\theta_j\}$ — неизвестные параметры, $t_1 = 0$, $t_2 = h$, $t_3 = 2h$, $\ldots$ — моменты измерений, $\{W_k\}$ — описанные выше ошибки наблюдений.

На одном графике изобразить: наблюдения в виде набора точек $\{(t_k, Y_k)\}$, истинную дальность $r(t) = \sqrt{x^2(t) + y^2(t)}$, ее МНК-оценку $\hat{r}(t) = \hat{\theta}_1 + \hat{\theta}_2 t$, а также доверительные границы $\hat{r}(t) \pm u_\alpha \sqrt{D_r(t)}$, где $(x(t), y(t))$ — истинные координаты самолета, $D_r(t)$ — дисперсия МНК-оценки $\hat{r}(t)$, а u_α — граница симметричного промежутка, в который стандартная нормальная величина попадает с вероятностью $1 - \alpha$.

Решение

1. Определим уравнение простой линейной регрессии (в матричной форме):

$$Y = A \cdot \theta + W$$

где:

- Y - вектор наблюдений (n x 1)
- A - регрессионная матрица (n x 2): первый столбец - единичный, второй - значения параметра t
- θ - вектор (2 x 1) неизвестных параметров (весов модели)
- W - матрица (n x 1) ошибок наблюдений
- n - количество наблюдений

$$A = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix};\ \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

1. Заметим что необходимое и достаточное условиене смещенности оценки выполняется, т.к. нет систематических ошибок:

$$MW = 0$$

2. Ошибки независимы, значит модель оптимальна:

$$I_n = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \Rightarrow DW_k = \sigma^2 = const$$

3. Оценим вектор параметров (весов) по методу наименьших квадратов:

$$\theta : (Y - A \cdot \theta)^2 = min_\theta$$

4. Если $\exists (A^T A)^{-1}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} (Y - A\theta)^2 = (A^T A)^{-1} A^T Y$$

5. Зададим доверительный интервал:

$$P(\hat{r}(t) - u_\alpha \sqrt{D_r(t)} \leq \hat{r}(t) + u_\alpha \sqrt{D_r(t)} = 1 - \alpha$$

Где:

- u_α - квантиль нормального распределения
- $D_r(t)$ - дисперсия МНК-оценки
- α - уровень значимости

1. Заметим что: $D_r(t) = M[(r - M[r])^2] \Rightarrow \hat{D}_r(t) = \frac{(Y - \hat{Y})^2}{n-2}$

Где:

- $\hat{Y} = A \cdot \hat{\theta}$ - оценка вектора наблюдений Y.

1. Выполним необходимые для решения задачи рассчеты, причем:

- `n` - колличество измерений
- `A` - регрессионная матрица A
- `Y` - вектор наблюдений Y
- `theta_hat` - МНК оценка параметров $\hat{\theta}$
- `theta_1_hat` - Первый параметр $\hat{\theta}_1$
- `theta_2_hat` - Второй параметр $\hat{\theta}_2$
- `r_hat` - МНК-оценки $\hat{r}$
- `s` - дисперсия МНК-оценки $D_r(t)$
- `upper_bound` , `lower_bound` - Границы доверительного интервала
- `E` - остатки (ошибка) МНК-оценки
- `u_a` - квантиль нормального распределения

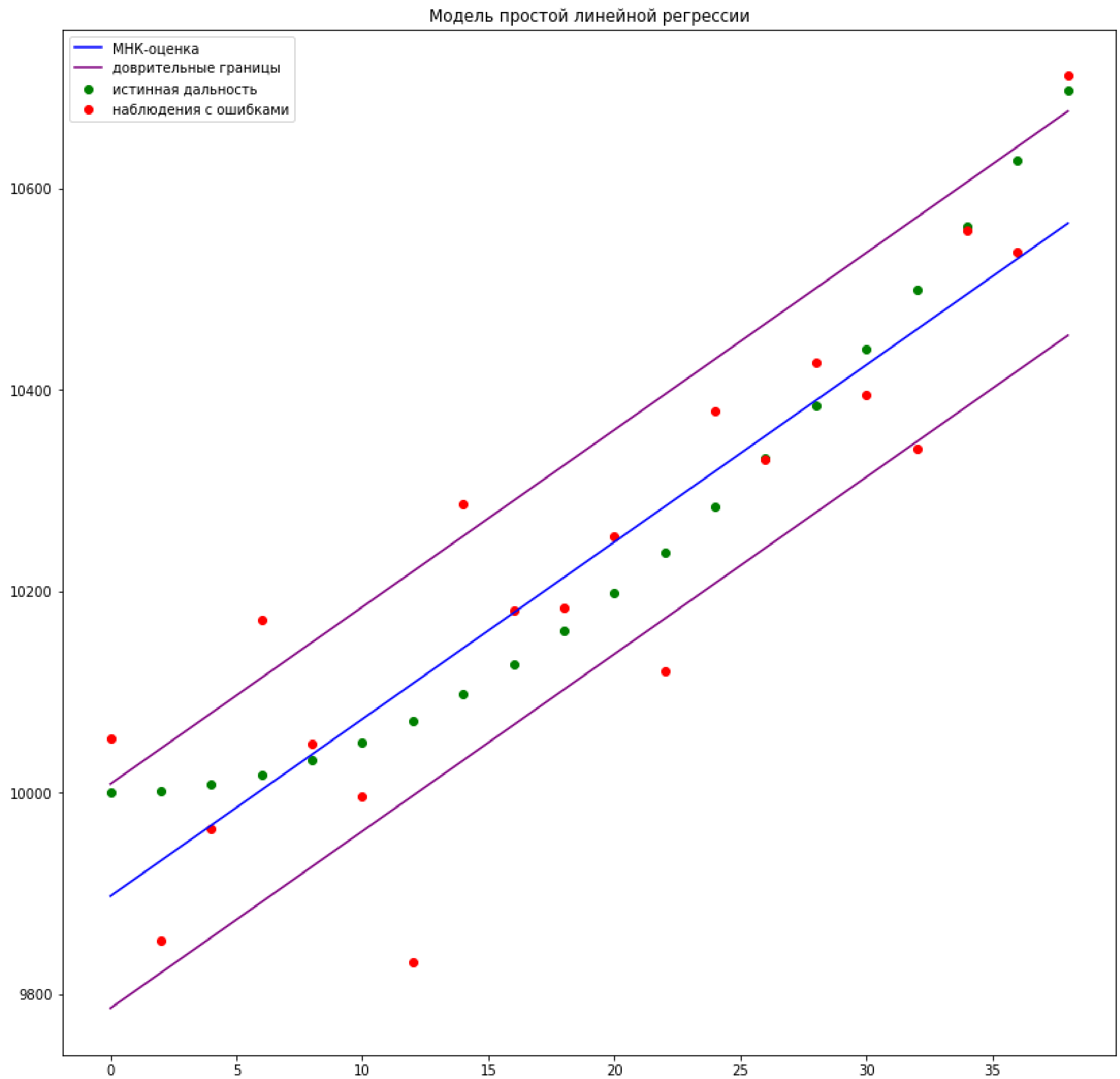
In [5]:

```
n = r.size
u_a = 2.326
A = np.matrix(np.concatenate((np.ones(shape = (n, 1), dtype = int), t.reshape(n,
Y = np.matrix(r+norm_errors).T
theta_hat = np.dot(np.dot(np.linalg.inv(np.dot(A.T,A)),A.T), Y)
theta_1_hat = float(theta_hat[0,0])
theta_2_hat = float(theta_hat[1,0])
```

```
r_hat = theta_1_hat + t * theta_2_hat
s = np.power(np.sum(r - r_hat),2)/(n-2)
upper_bound = r_hat + u_a*np.sqrt(s)
lower_bound = r_hat - u_a*np.sqrt(s)
E = r+norm_errors - r_hat
```

1. Построим график

In [6]:
```
plt.figure(figsize=(14,14))
plt.title("Модель простой линейной регрессии")
plt.scatter(t,r, label = "истинная дальность", color = 'green')
plt.scatter(t,r+norm_errors, label = "наблюдения с ошибками", color = 'red')
plt.plot(t, r_hat, label = "МНК-оценка", color = 'blue')
plt.plot(t, upper_bound, label = "доврительные границы", color = 'purple')
plt.plot(t, lower_bound, color = 'purple')
plt.legend()
plt.show()
```



Задача 3

3. Найти МНК-оценки параметров и дальности, считая, что наблюдения описываются моделью параболической линейной регрессии

$$Y_k = \theta_1 + \theta_2 t_k + \theta_3 t_k^2/2 + W_k, \quad k = 1, \ldots, n.$$

Представить графические данные (по аналогии с предыдущим пунктом).

Решение

1. Решение аналошично предыдущему заданию с той лишб разницей что матрица A и θ теперь имееют размер (n x 3) и (3 x 1) соответственно:

$$A = \begin{bmatrix} 1 & t_1 & \frac{t_1^2}{2} \\ \vdots & \vdots & \vdots \\ 1 & t_n & \frac{t_n^2}{2} \end{bmatrix}; \; \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

1. Выполним необходимые для решения задачи рассчеты, причем:

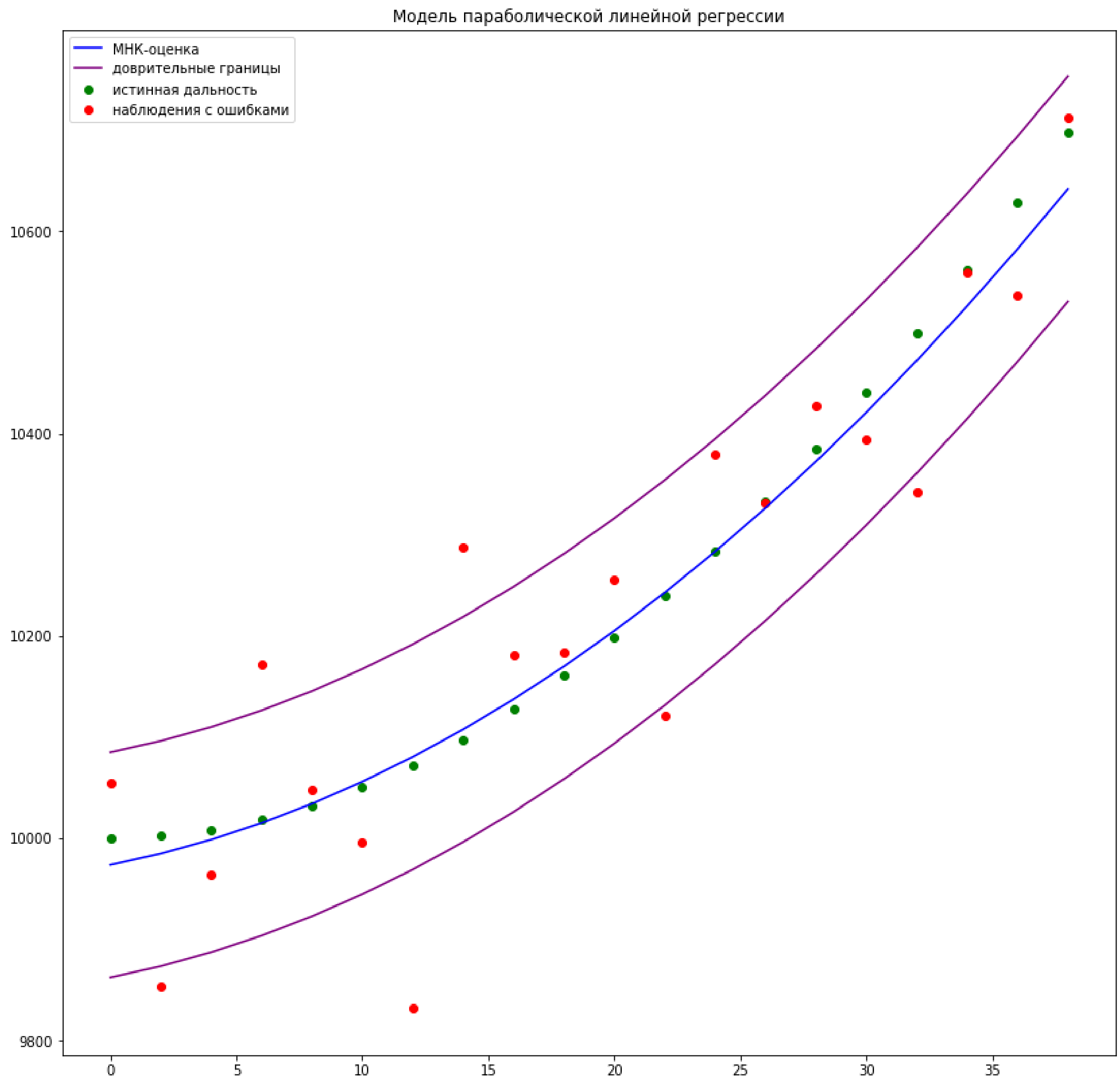
- `n` - колличество измерений
- `A` - регрессионная матрица A
- `Y` - вектор наблюдений Y
- `theta_hat` - МНК оценка параметров $\hat{\theta}$
- `theta_1_hat` - Первый параметр $\hat{\theta}_1$
- `theta_2_hat` - Второй параметр $\hat{\theta}_2$
- `theta_3_hat` - Третий параметр $\hat{\theta}_3$
- `r_hat` - МНК-оценки $\hat{r}$
- `s` - дисперсия МНК-оценки $D_r(t)$
- `upper_bound` , `lower_bound` - Границы доверительного интервала
- `E2` - остатки (ошибка) МНК-оценки
- `u_a` - квантиль нормального распределени

In [7]:
```
n = r.size
A = np.matrix(np.concatenate((np.concatenate((np.ones(shape = (n, 1), dtype = ir
Y = np.matrix(r+norm_errors).T
theta_hat = np.dot(np.dot(np.linalg.inv(np.dot(A.T,A)),A.T), Y)
theta_1_hat = float(theta_hat[0,0])
theta_2_hat = float(theta_hat[1,0])
theta_3_hat = float(theta_hat[2,0])
r_hat = theta_1_hat + t * theta_2_hat + np.power(t,2) * theta_3_hat/2
s = np.power(np.sum(r-r_hat),2)/(n-2)
upper_bound = r_hat + u_a*np.sqrt(s)
lower_bound = r_hat - u_a*np.sqrt(s)
E2 = r+norm_errors - r_hat
```

1. Построим график

In [8]:
```
plt.figure(figsize=(14,14))
plt.title("Модель параболической линейной регрессии")
plt.scatter(t,r, label = "истинная дальность", color = 'green')
```

```
plt.scatter(t,r+norm_errors, label = "наблюдения с ошибками", color = 'red')
plt.plot(t, r_hat, label = "МНК-оценка", color = 'blue')
plt.plot(t, upper_bound, label = "доврительные границы", color = 'purple')
plt.plot(t, lower_bound, color = 'purple')
plt.legend()
plt.show()
```



Задача 4

4. На уровне значимости α для модели из предыдущего пункта проверить гипотезу о том, что $\theta_3 = 0$.

Решение

1. Введем нулевую гепотизу H_0: $\theta_3 = 0$
2. Тогда альтернативная гипотиза H_1: $\theta_3 \neq 0$

3. Введем статистику: $T = \frac{\hat{\theta}_3}{\sqrt{D[\hat{\theta}_3]}}$
4. Критическая область: $P(-u_{1-\frac{\alpha}{2}} \leq T \leq u_{1-\frac{\alpha}{2}}) = 1 - \alpha$
5. Из свойств теоремы Гаусса-Маркова следует: $M[\hat{\theta}_3] = \theta_3$
6. Произведем рычисления для данной задачи:

$$T \mid_{H_0} = \frac{\hat{\theta}_3}{\sqrt{(\hat{\theta}_3 - 0)^2}}; \ \alpha = 0.01$$

$$P(-u_{0.995} \leq \frac{\hat{\theta}_3}{\sqrt{(\hat{\theta}_3 - 0)^2}} \leq u_{0.995}) = 0.99 \Rightarrow P(-2.58 \leq 1 \leq 2.58)$$

При выполнении гипотезы H_0, статистика T входит в критическую область, следовательно она верна на уровне значимости $\alpha = 0.01$. Значит $\theta_3 = 0$.

Задача 5

5. Для каждой их двух моделей построить остатки. По вектору остатков построить гистограмму. Изобразить полученные гистограммы на одном графике с плотностью $\mathcal{N}(0, \sigma^2)$. Можно ли по этому графику сделать вывод о правильности выбора модели?

Решение

1. Построим график остатков модели простой линейной регрессии с помощью `plt.scatter(t,E)`
2. С помощью `scipy.stats.norm.pdf(x)` построим график плотности $N(0, \sigma^2)$
3. С помощью plt.hist построим на том же графике гистограмму по вектору остатков

In [21]:
```
import scipy.stats as stats
plt.figure(figsize=(14,14))
plt.scatter(t,E)
plt.title("График остатков модели простой линейной регрессии")
plt.show()

x = np.linspace(stats.norm.ppf(0.01),
                stats.norm.ppf(0.99), 100)

plt.figure(figsize=(14,14))
plt.plot(x, stats.norm.pdf(x))
plt.hist(E/100, density=True, histtype='stepfilled', alpha=0.5)
plt.title("Гистограмма остатков и график плотности $N(0, \sigma^2)$")
plt.show()
```

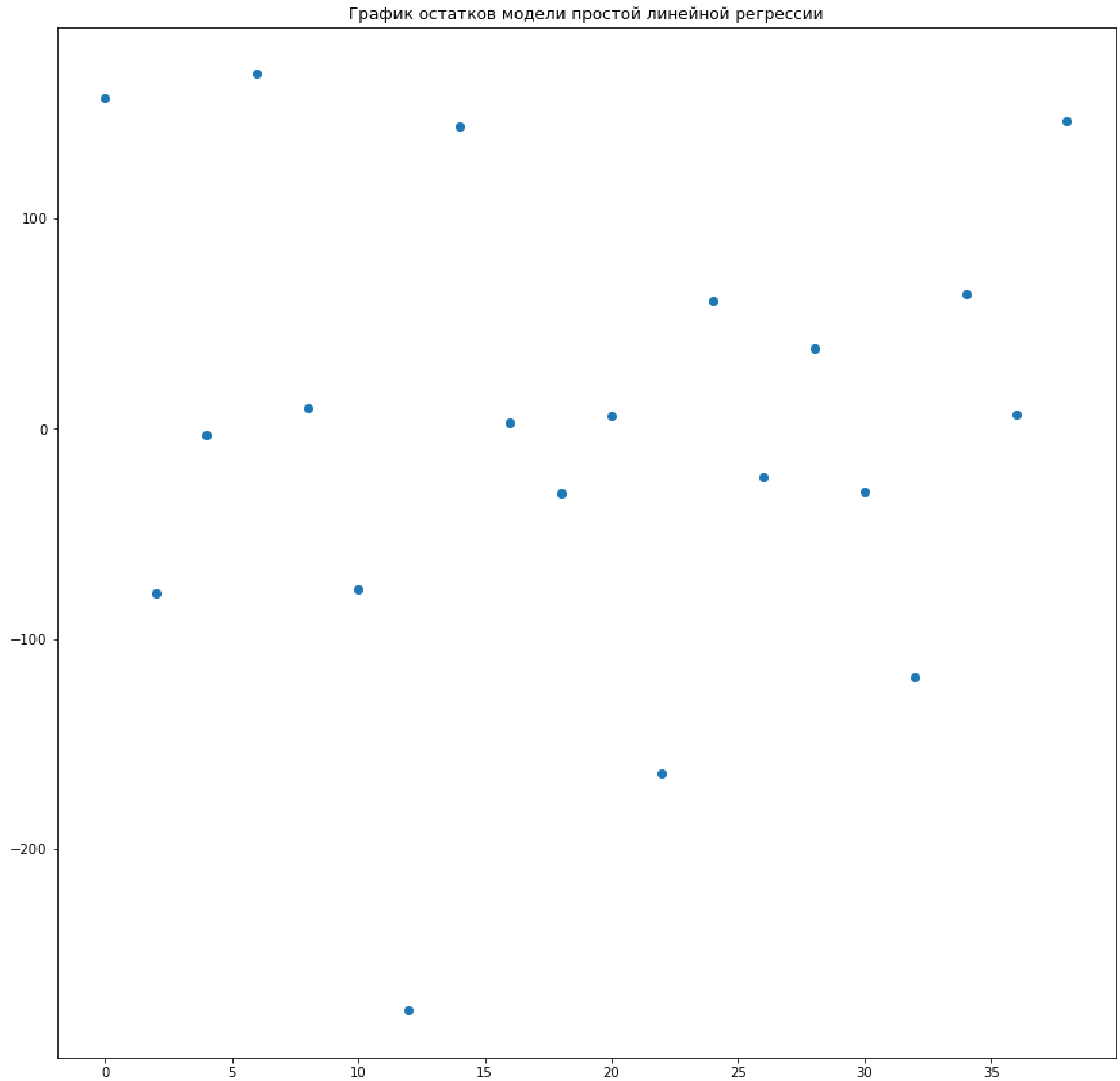
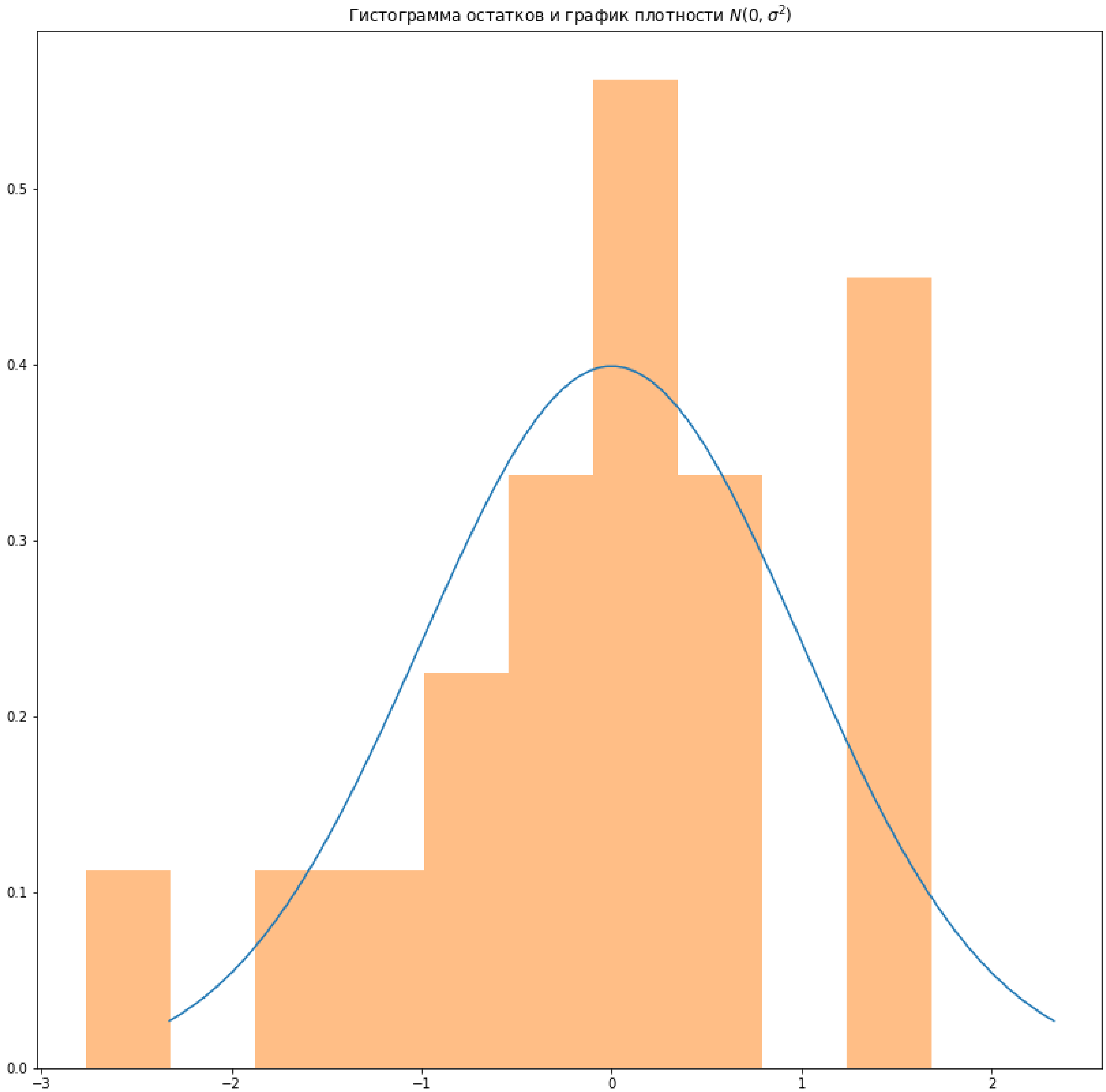


График остатков модели простой линейной регрессии
100
0
−100
−200
0
5
10
15
20
25
30
35



1. Аналогично и для случая параболической линейной регрессии

In [22]:
```
plt.figure(figsize=(14,14))
plt.scatter(t, E2)
plt.title("График остатков модели параболической простой линейной регрессии")
plt.show()

#
x = np.linspace(stats.norm.ppf(0.01),
                stats.norm.ppf(0.99), 100)

plt.figure(figsize=(14,14))
plt.plot(x, stats.norm.pdf(x))
plt.hist(E2/100, density=True, histtype='stepfilled', alpha=0.5)
plt.title("Гистограмма остатков и график плотности $N(0, \sigma^2)$")
plt.show()
```

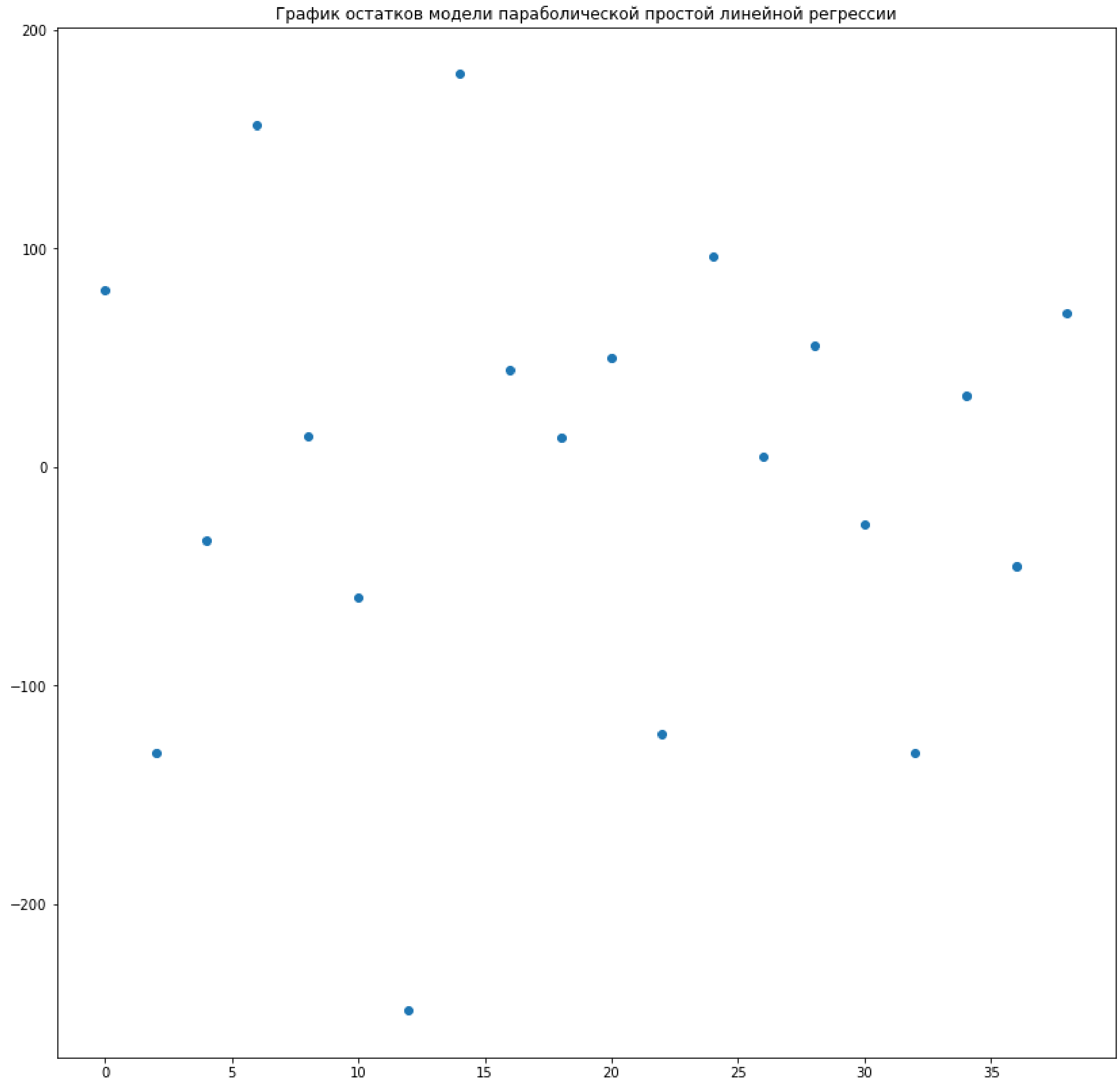
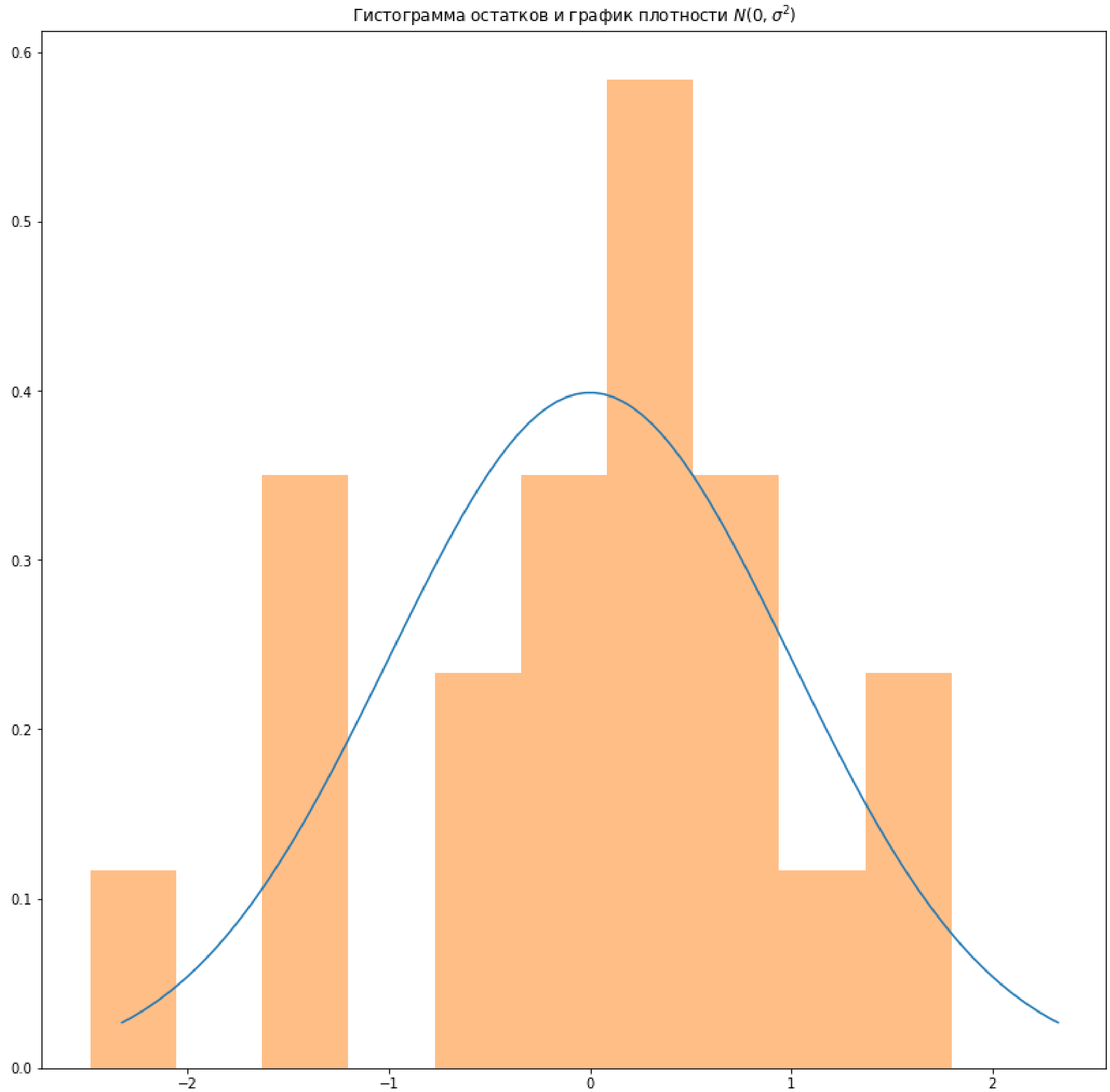


График остатков модели параболической простой линейной регрессии
200
100
0
−100
−200
0
5
10
15
20
25
30
35



1. Из полученных графиков видно, что следует выбрать модель простой линейной регрессии. Это утверждение также подтверждается в задании 4 - гипотеза H_0 верна на уровне значимости $\alpha = 0.01$.